JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY

4/2025

vol. 102

(Contents continued on back cover)

eISSN 1899-8852

Aware Node Localization in Wireless Sensor Networks Using Harris Hawks Optimization

Seddik Rabhi

University of Adrar, Adrar, Algeria

https://doi.org/10.26636/jtit.2025.4.2285

Abstract — Precise and efficient localization is a key enabler for context-aware operations in emerging 6G cognitive semantic communication (CSC) systems. In AI-native and semantic-aware networks, precise node positioning improves semantic compression, context-driven routing, and adaptive spectrum allocation, positively affecting communication reliability and resource utilization efficiency. This paper addresses the problem of localization in wireless sensor networks (WSNs) in the broader context of 6G CSC, formulating it as an optimization task. Based on the previous research, we explore the application of bio-inspired metaheuristic algorithms to achieve robust and high accuracy positioning. Specifically, we propose the use of the Harris hawks optimization (HHO) algorithm to develop a semantic-aware, stable, and efficient localization framework. The proposed approach is implemented and tested within the Matlab simulation environment. Performance evaluation is conducted through comparative experiments with two widely used optimization algorithms: particle swarm optimization (PSO) and cuckoo search optimization (CSO). The simulation results demonstrate that the proposed HHO-based localization method not only improves positioning accuracy by up to 25% compared to the benchmarks, but also provides enhanced stability, enabling its integration with CSC architectures for intelligent resource management in next-generation networks.

Keywords — 6G cognitive semantic communication, context-aware resource management, Harris hawks optimization, node positioning, semantic-aware localization, WSN

1. Introduction

Wireless sensor networks (WSNs) are composed of a large number of sensor nodes which are densely distributed and interconnected via a wireless medium. Their primary mission is to collect and transmit environmental information in real time to support the observation and monitoring of various physical phenomena, such as meteorological data, health status, as well as security- and surveillance-related parameters. In the context of 6G cognitive semantic communication (CSC) systems, precise node localization becomes even more important, as spatial context directly enhances semantic-aware data processing, context-driven routing, and intelligent spectrum allocation [1], [2].

Installation of GPS receiver sensors on each sensor node is often impractical and costly. Therefore, alternative localization methods have been developed under the assumption that only a subset of nodes, called anchors, is equipped with GPS and knows its exact position [3]. In 6G WSNs, localization also plays a role in reliable and efficient communication, where intelligent transmission strategies rely on positional knowledge to optimize the lifetime of the network and improve data relevance [4].

Over the last decade, optimization-based localization methods have received much attention due to their ability to improve sensor positioning accuracy and reduce estimation errors. These methods reformulate localization as an optimization problem which requires the definition of an objective function [5].

The localization process generally consists of two stages: estimation of distance between sensors and subsequent calculation of their position. Recent research favors nature-inspired metaheuristic algorithms to address localization challenges. These algorithms randomly generate an initial solution and iteratively refine it by optimizing the difference between the measured distance from an unknown node to the anchors and the Euclidean distance computed from the estimated node position and anchor coordinates.

The first metaheuristic approach applied to localization was simulated annealing (SA) [6], followed by particle swarm optimization (PSO) [7] and genetic algorithms (GA) [8]. These early successes spurred extensive experimentation with other metaheuristics, including the chicken swarm optimization algorithm (CSO) [9] and cuckoo search optimization (CSO) [10]. The authors of [11] introduced two localization approaches based on the fruit fly optimization algorithm (FOA). These studies collectively demonstrated the adaptability of metaheuristics to WSN localization, making them suitable candidates for integration into next-generation semantic-aware, energy-efficient, and context-driven 6G WSN architectures.

This study addresses the problem of accurate node localization in wireless sensor networks (WSNs) by introducing a novel localization framework based on the Harris hawks optimization (HHO) algorithm. Taking advantage of the dynamic balance between exploration and exploitation offered by HHO, the proposed method, termed HHO-L, formulates the localization task as a non-linear optimization problem, minimizing the discrepancy between the estimated and actual distances from the anchor nodes.

The main contributions of this work can be summarized as follows.

- Novel application of HHO to WSN localization. This is among the first attempts to adapt Harris hawks optimization to the node localization problem, demonstrating its potential in handling complex and high-dimensional search spaces.
- Robust formulation of the objective function. The localization problem is modeled to minimize the localization error by integrating the estimated distances from the signal strength indication (RSSI), the time of arrival (TOA), and other range-based metrics with Euclidean distance calculations.
- Performance benchmarking. A comprehensive comparative analysis is conducted against two well-established metaheuristics, particle swarm optimization (PSO) and cuckoo search optimization (CSO), to evaluate localization accuracy, convergence behavior, and stability.
- Simulation-driven validation. Matlab-based experiments are performed under varying network densities, anchor ratios, and deployment scenarios to prove the accuracy and robustness of the approach.
- Alignment with next-generation networks. The study situates HHO-L within the context of emerging 6G cognitive semantic communication (CSC) paradigms, highlighting its potential for integration into location-aware, metaheuristic-driven network optimization frameworks.

The remainder of this article is structured as follows. Section 2 presents an overview of localization in WSNs, reviews the related literature, and highlights the limitations of existing methods. Section 3 describes the Harris hawks optimization algorithm and details the proposed HHO-L localization framework, including its mathematical formulation and implementation steps. Section 4 reports and discusses the experimental results, including a comparative evaluation with PSO and CSO. Finally, Section 5 concludes the paper and outlines potential directions for future research.

2. Related Works

Localization in WSNs has been studied over the past two decades, as it plays a crucial role in enabling context-aware operations such as routing, monitoring, and environmental sensing. Several techniques have been proposed to estimate the positions of unknown nodes, ranging from traditional range-based and range-free methods to more advanced optimization-based approaches. An excellent review of this topic is presented in article [12].

Among the most well-known metaheuristics applied to the localization problem is PSO [7], which simulates the social behavior of flocks of birds. PSO has been widely adopted due to its simplicity and fast convergence. However, it often suffers from premature convergence and becomes trapped in local optima when dealing with complex, high-dimensional search spaces. To overcome these limitations, more recent approaches have turned to nature-inspired algorithms with stronger exploration and exploitation capabilities.

Cuckoo search optimization (CSO) is one such algorithm that mimics the behavior of cuckoo birds. CSO has shown promising results in WSN localization [10], offering better performance than PSO in several studies due to its Lévy flight-based search mechanism which enhances global exploration. However, while CSO improves convergence and avoids local optima more effectively, it can still exhibit instability under specific deployment scenarios or sparse anchor configurations.

More recently, bio-inspired metaheuristics, such as bat algorithm, whale optimization algorithm (WOA), and gray wolf optimizer (GWO) have also been adapted for node localization [13], with varying degrees of success. These algorithms aim to balance the trade-off between exploration and exploitation by simulating specific natural behaviors like echolocation or pack hunting. While these techniques offer improved robustness and adaptability, they still face challenges in terms of accuracy, convergence speed, and sensitivity to parameter tuning.

In addition to single-algorithm approaches, hybrid and improved localization strategies have emerged. For example, the authors of [14] proposed a regularized least squares DV hop method to enhance multihop localization accuracy, while in [15], DV hop using RSSI-based distance estimation and recursive computation, significantly reducing localization errors, was improved. In parallel, UAV-assisted approaches have been explored to enhance network connectivity and reduce localization error in large-scale or complex deployments.

In [16], an energy efficient UAV flight path model with metaheuristic optimization for cluster head selection in next generation WSNs was proposed, demonstrating the benefits of UAV mobility for improving network lifetime and coverage. Similarly, a Java macaque algorithm for optimizing VANET routes based on the IoT was presented in [17], illustrating the adaptability of meta-heuristic frameworks to various wireless network contexts, including localization.

In this context, the Harris hawks optimization (HHO) algorithm has emerged as a novel and powerful optimization tool [18]. HHO mimics the cooperative hunting behavior of Harris' hawks and dynamically adjusts its search patterns between soft and hard besiege strategies. Although HHO has shown success in various domains such as feature selection, machine learning, and engineering design, its application to WSN localization remains relatively underexplored. The current study contributes to bridging this gap by adapting and evaluating HHO for node localization and comparing its performance against PSO and CSO under various network configurations.

Beyond traditional WSNs, recent research has highlighted the importance of accurate localization in the emerging field concerned with 6G cognitive semantic communication (CSC).

In AI-native 6G networks, precise node localization is not merely a positioning task, but a foundational enabler for semantic-aware and context-driven communication. Accurate

location information supports semantic compression by providing spatial context that determines which sensing data are relevant, thereby reducing unnecessary transmissions and improving network efficiency in applications such as V2X, UAV swarms, and IoT deployments for the metaverse [19]–[21].

The authors of [22] analyzed localization performance using a channel knowledge map (CKM) in a 3D environment, integrating angle-of-arrival, angle-of-departure, and path-loss information to achieve submeter accuracy. Their CRLB-based analysis demonstrated the impact of propagation paths and grid resolution on positioning performance, offering valuable information on optimizing localization accuracy in 6G communication contexts.

Similarly, in [23], a reconfigurable intelligent surface (RIS) system assisted by UAVs was proposed for vehicle positioning in dense urban environments, using the snake optimization algorithm to dynamically adjust the placement of RIS. This work illustrates how metaheuristic-driven localization can be embedded in integrated sensing and communication (ISAC) frameworks, directly aligning with the 6G CSC goals for intelligent transportation and dynamic network optimization.

Localization is also crucial for intelligent routing and dynamic spectrum allocation in CSC, where location-aware decisions improve robustness, reduce latency, and enhance resource utilization [20], [24], [25]. In advanced 6G infrastructures, such as those leveraging intelligent surfaces or edge-based semantic processing, localization enables adaptive beamforming and spectrum reconfiguration to maintain high-quality links [20], [24]. Moreover, semantic-aware frameworks increasingly integrate location data into multimodal resource prioritization strategies for scenarios such as smart traffic systems, immersive environments, and mission critical industrial IoT [26]–[28].

Parallel to these developments, AI and meta-heuristic algorithms are playing a central role in multi-task/multimodal optimization for CSC. Machine learning, reinforcement learning, and evolutionary optimization have been applied to semantic spectrum allocation, channel selection, and joint localization–communication design, allowing adaptive and context-aware resource management in highly dynamic 6G environments [19], [20], [29], [30].

Metaheuristics such as HHO, GA, and swarm intelligence methods offer the advantage of handling high-dimensional optimization problems with non-convex constraints, making them suitable for integrated CSC tasks involving spectrum management, edge computing, and secure semantic data delivery [31]–[33]. Joint optimization frameworks that combine localization with semantic-aware transmission have been shown to improve end-to-end performance under latency, utility, and security constraints [33]–[35].

This body of work, including recent advances in CKM-based positioning [22] and UAV-assisted RIS metaheuristic optimization [23], suggests that integrating high-accuracy localization algorithms into CSC frameworks could directly improve semantic compression efficiency, adaptive spectrum allocation, and network resilience.

2.1. Research Gap

Despite significant progress in WSN localization, several challenges remain, particularly when positioning is viewed through the lens of next-generation wireless networks and CSC paradigms. Traditional range-based and range-free algorithms, as well as classical metaheuristics such as PSO and CSO, have demonstrated effectiveness in moderate-scale, relatively stable network conditions. However, these methods often struggle with:

- Balancing exploration and exploitation in high-dimensional search spaces. Many existing algorithms converge prematurely, becoming trapped in local optima, which limits their accuracy in complex and irregular deployments.
- Robustness under dynamic and sparse-anchor conditions. Localization accuracy tends to degrade significantly when anchor nodes are sparse, network topology changes, or environmental noise increases.
- Integration into next-generation semantic-aware networks. Most existing localization studies focus on accuracy within traditional WSN deployments and overlook integration into AI-native, 6G ready infrastructures where location data must also serve real-time semantic compression, context-driven resource allocation, and cross-layer optimization.
- Underexplored potential of advanced metaheuristics. Although bio-inspired algorithms such as bat algorithm, GWO, and WOA have been applied to WSN localization, newer approaches like HHO, with adaptive transition strategies between exploration and exploitation, have not been extensively evaluated.

Given these limitations, there is a clear need for a localization method that not only improves accuracy and stability over existing metaheuristics, but also aligns with the multi-objective demands, where node positioning directly impacts semantic communication efficiency, spectrum utilization, and network resilience.

The proposed HHO-based localization framework addresses this gap by combining high-dimensional optimization capability with adaptability to varying deployment conditions, making it a candidate for integration into future intelligent and semantic-aware WSN architectures.

3. Methodology

Harris hawks optimization (HHO) is a nature-inspired metaheuristic optimization algorithm that simulates the cooperative hunting behavior of Harris' hawks, a predatory bird species known for their intelligent and collaborative strategies in capturing prey. Originally introduced in [18], HHO mimics the social hierarchy and dynamic tactics employed by these raptors, such as surprise pounce and perching strategies, to explore and exploit the search space effectively. The algorithm adaptively balances exploration and exploitation phases, making it suitable for solving complex and non-linear optimization problems.

The HHO algorithm begins with an initial population of candidate solutions, represented as a set of individuals. These individuals are considered "hawks" in the algorithm. Each hawk represents a potential solution to the optimization problem.

The algorithm iteratively updates the positions of the hawks in search of better solutions. It employs various operators inspired by the behavior of Harris' hawks, such as exploration, exploitation, and flight. These operators allow the hawks to explore the search space, exploit promising regions, and escape from local optima.

During the optimization process, the hawks communicate and cooperate with each other to improve the overall performance. They exchange information, share knowledge, and learn from their experiences to guide the search toward optimal solutions. The main objective of the HHO algorithm is to find the best solution that optimizes a given fitness function. This approach can be applied to a wide range of optimization problems, including engineering design, planning, data mining, and many others [18].

Exploration is the initial phase of the hunting process, which involves observing, tracking, and locating the prey. In the context of the HHO algorithm, this step is referred to as the exploratory mechanism. In nature, Harris' hawks may spend several hours searching for prey. Similarly, in the algorithm, the probability of spotting the prey (target solution) depends on the quality of the hawks (candidate solutions). Therefore, the best candidate solution is the one closest to the prey.

Harris' hawks adopt two strategies while waiting for their prey. They either position themselves near other family members to initiate a group attack, or they choose random vantage points such as high trees. Both strategies are modeled and represented by Eq. (1).

$$\begin{aligned} &X(t+1) = \\ &X_{rand}(t) - r_1\left|X_{rand}(t) - 2\,r_2 X(t)\right| \;\text{ for }\; q \geqslant 0.5 \\ &X_{rabbit}(t) - X_m(t) - r_3\big(LB + r_4(UB - LB)\big) \text{ for } q < 0.5 \end{aligned} \tag{1}$$

where $X(t+1)$ represents the new positions of the hawks at iteration t, X_{prey} corresponds to the location of prey, denotes a hawk selected randomly in the search space, and $X\ (t)$ indicates the initial locations of the hawks, which are calculated according to Eq. (2). Variables r_2, r_3, r_4, r_5 and q are random values between 0 and 1. It is important to note that these random numbers are updated at each iteration t.

$$X_m(t) = \frac{1}{N}\sum_{i=1}^{N} X_i(t)\ , \tag{2}$$

where X_i represents the position of the hawk in iteration t and N is the total number of hawks.

The HHO algorithm can move from the exploration phase to the exploitation phase by adapting its exploitation behaviors according to the escape energy of the prey. During the escape phase, the energy of prey decreases significantly. To take this reduction into account, the energy of prey is modeled as follows:

$$E = 2\,E_0\left(1 - \frac{t}{T}\right)\ . \tag{3}$$

After discovering the prey during exploration (i.e. extended search), the next phase marks the start of exploitation. Harris' hawks then attempt to swoop down on their prey suddenly. On the other hand, the prey attempts to escape, which is commonly called the seven escapes.

Optimization of Harris' hawks proposed four potential approaches for modeling hunting strategies and escape behaviors. A random number r is used to represent the probability of success of the prey in the fight ($r < 0.5$) or its failure ($r \geqslant 0.5$). Additionally, Harris' hawks use either a soft block or a hard block to capture the prey, depending on the strength of the prey E. For example, if the block is soft, the condition will be $E \geqslant 0.5$, otherwise $E < 0.5$.

In the soft siege phase, when the values of E are greater than or equal to 0.5 with r greater than or equal to 0.5, this means that the prey has sufficient energy to defend itself against Harris' hawks by following random paths and performing deceptive jumps. Unfortunately, the prey will fail because Harris' hawks exhaust their energy by circling them and then launching a surprise attack. Eq. (4) shows the modeling of this behavior.

$$\begin{aligned} X(t+1) =& \Delta X(t) - E\left|JX_{rabbit}(t) - X(t)\right| \text{ or} \\ &\Delta X(t) = X_{rabbit}(t) - X(t)\ , \end{aligned} \tag{4}$$

$$J = 2 \times (1 - r_6)\ . \tag{5}$$

In the soft siege phase, X represents the difference in position between the prey and their initial position in iteration t. The value r_6 is chosen randomly within the interval $0 \ldots 1$. For J, it refers to the random jump of the prey, and its value changes randomly to imitate the nature of the prey's movements.

In the hard phase $t < 0.5$ and $r < 0.5$. Therefore, the pray does not have enough energy to escape. In addition, Harris's hawks are ready to surround the prey and carry out a surprise attack, with difficulty. Equation 6 illustrates the update of current positions in this situation.

$$X(t+1) = X_{prey}(t) - E\left|\Delta X(t)\right|\ . \tag{6}$$

The third case, soft seat with progressive rapid dips, is more complicated than the one described above, because it is used when $|E| \geqslant 0.5$ and $r < 0.5$. Thus, the prey has enough power to escape successfully. On the other hand, Harris' hawks continue to perform numerous rapid dives to force the prey to change trajectory and distract it. The process continues until the best time to catch the prey is chosen. The following equation describes the decision to move to implement soft encirclement.

$$Y = X_{prey}(t) - E\left|JX_{prey}(t) - X_{(t)}\right|\ . \tag{7}$$

If Harris' hawks notice that the prey is making deceptive movements and is about to escape, they will intensify their sharp, irregular, and rapid dives. The new technique of the hawks is based on the Levy fights (LF) as:

$$Z = Y + S \times LF(D)\ . \tag{8}$$

where D indicates the dimension of the problem, S refers to a random vector of size $1 \times D$, and LF is calculated according to:

$$LF(x) = 0.01 \, \frac{u \, \sigma}{|v|^{\frac{1}{\beta}}} \, , \tag{9}$$

$$\sigma = \frac{\Gamma(1+\beta) \, \sin(\frac{\pi\beta}{2})}{\Gamma \, \frac{1+\beta}{2} \, \beta \, 2^{\frac{\beta-1}{2}}} \, , \tag{10}$$

where u and v refer to a random value in the interval $0 \ldots 1$, β indicates the fixed variable defined in Eq. (10).

Therefore, the mathematical model for updating the positions of hawks in the soft circling stage is given by the following equation.

$$X(t+1) = \begin{cases} Y \;\; \text{for } F(Y) < F\big(X(t)\big) \\ Z \;\; \text{for } F(Z) < F\big(X(t)\big) \end{cases} . \tag{11}$$

In the case given by Eq. (11), Y represents the value calculated according to the specified formula, while Z represents the value calculated according to Eq. (8).

The rigid seat with progressive rapid dips is the last case, when the values of $r < 0.5$ and $|E| < 0.5$. This means that the prey does not have enough strength to escape. At the same time, hawks seek to reduce the space between themselves and the prey before surprising and attacking it. Equation (12) describes the updating of the hawks' positions.

$$Y = X_{prey}(t) - E\big|JX_{prey}(t) - X_m(t)\big| \, . \tag{12}$$

The summary of the HHO algorithm-based procedures is presented in pseudo-code form as Algorithm 1, where the value of $X_m(t)$ is calculated using Eq. (2).

3.1. Proposed Algorithm

The goal of the localization process in WSNs is to calculate the coordinates of N unknown sensors based on the known position of sensors. It is assumed that all sensors are deployed in a two-dimensional area of interest, all sensors have similar hardware characteristics, particularly in terms of their ability to send and receive information (similar connectivity radius R), and that each sensor is equipped with a similar radio interrogator which allows sensors in the neighborhood to estimate the distance between them.

The localization process based on the Harris hawks optimization algorithm is illustrated as Algorithm 1.

Furthermore, localization consists in calculating the coordinates of unknown nodes (target nodes) using the inter-node distance information managed by the anchors.

The basic steps of the positioning process are described below.

1. The network sensors (N known nodes and M unknown nodes) are randomly deployed within the area of interest.

2. Known nodes (considered anchors during this process) frequently broadcast their positions until the localization process is completed.

3. Each unknown node receives an RSSI radio signal through three or more anchors, so it is considered a localizable sensor.

Algorithm 1 Harris hawks optimization (HHO) [18]

Inputs: The population size N and maximum number of iterations T
Outputs: The location of rabbit and its fitness value
Initialize the random population $X_i, i = 1, 2, \ldots, N$
while stopping condition is not met **do**
  Calculate fitness values of hawks
  Set X_{rabbit} as the location of rabbit (best location)
  for each hawk X_i **do**
    Update the initial energy E_0 and jump strength J
    $E_0 = 2\,\text{rand}() - 1, \;\; J = 2\,(1 - \text{rand}())$
    Update the E using Eq. (3)
    if $|E| \geqslant 1$ **then**
      Update the location vector using Eq. (1)
    end if
    if $|E| < 1$ and $r \geqslant 0.5$ and $|E| \geqslant 0.5$ **then**
      Update the location vector using Eq. (4)
    else if $r \geqslant 0.5$ and $|E| < 0.5$ **then**
      Update the location vector using Eq. (6)
    else if $r < 0.5$ and $|E| \geqslant 0.5$ **then**
      Update the location vector using Eq. (11)
    else if $r < 0.5$ and $|E| < 0.5$ **then**
      Update the location vector using Eq. (12)
    end if
  end for
end while
Return X_{rabbit}

Furthermore, we assume that the distance measurement between neighboring captures can then be transferred to equivalent distances based on the RSSI technique.

The real distance d_{ij} between target node x_i, y_i and the j-th anchor x_j, y_j is defined by the following Euclidean distance as:

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \, . \tag{13}$$

The measured distance d_{ij} between target node x_i, y_i and the j-th anchor x_j, y_j is formulated as follows:

$$\hat{d}_{ij} = [d_{ij} \pm n_{ij}] \, , \tag{14}$$

where n_{ij} represents the error between target node x_i, y_i, and its neighboring anchor x_j, y_j.

4. The objective function of the localization problem represents the mean-square difference between a target node and the corresponding anchors. It is formulated by the following equation:

$$f(x,y) = \frac{1}{N} \sum_{i=1}^{N} \left(\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} - \hat{d}_{ij} \right)^2 , \tag{15}$$

where N ($N \geqslant 3$) represents the number of anchors in the unknown communication range, the target node can calculate its geographic coordinates by running the HHO algorithm, represented by the flow diagram given in Fig. 1.

The objective function $f(x_i, y_i)$ is minimized, then (x_i, y_i) represents the coordinate of the ith localizable node.

5. The sensors located during this process act as anchors during the next iterations, so that the number of anchors

increases with the iterations and the number of unknown nodes decreases. But this will only be used to locate nodes initially considered non-localizable, i.e. those that do not have three or more anchors in their transmission range.

6. Steps 2 to 5 are repeated until all unknown nodes are localized.

7. Now we can calculate the average localization error as the average Euclidean distance between the real positions and the estimated coordinates of all unknown sensors. Thus, the average localization error can be calculated using the following equation:

$$E_L = \frac{1}{N_L} \sum \sqrt{(X_{real} - X_{rabbit})^2 - (Y_{real} - Y_{rabbit})^2}\,, \tag{16}$$

where N_L is the number of unknown nodes, X_{rabbit}, Y_{rabbit} are the coordinates of the calculated node and X_{real}, Y_{real} is the position of the real node, $L = M - N$, M is the number of unknown nodes, and N is the number of anchors.

The smaller the average error, the higher the localization accuracy. Therefore, the challenge is to reduce the average error, and in this way to transform the localization problem into an optimization problem.

4. Results

In this section, the performance of the proposed HHO-L optimization algorithm was evaluated based on the density of anchor nodes, the density of unknown nodes, communication range and population size.

At the same time, a comparative study was carried out between the proposed approach and two recent similar works which use one of the most popular metaheuristic algorithms: PSO [7] and CSO [9], noting that comparative tests are performed using the same network configuration.

Simulations are performed using the Matlab environment, with their parameters summarized in Tab. 1.

To facilitate parameter modification and to generate multiple results for evaluating the performance of our proposed approach, a simulation interface was developed for flexible experimentation and analysis. It is illustrated in Fig. 2.

The large white square in the interface shows the deployment of nodes within a two-dimensional area. The orange lozenges represent anchor nodes, while the black squares correspond to unknown nodes. The blue circles indicate the estimated positions of the unknown nodes. The network parameters selected for this deployment are presented in the side panels, shown in the small boxes adjacent to the simulation area.

Tab. 1. Configuration of WSN during simulation.

Parameter	Value
Number of sensor nodes	30 – 60
Number of anchors nodes	5 – 20
Deployment area	50 m × 50 m
Transmission range	$\sum_{i=6}^{i=12} 5 \times i$ [m]
Maximum iterations	10 – 30
Population size	15 – 30

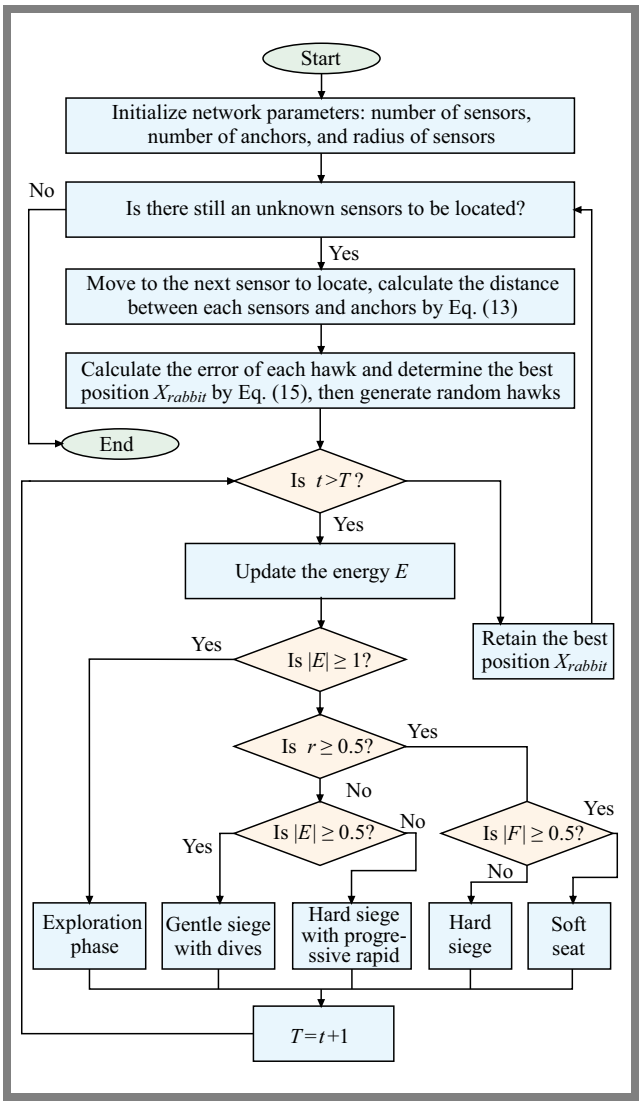

Fig. 1. Node localization by proposed method.

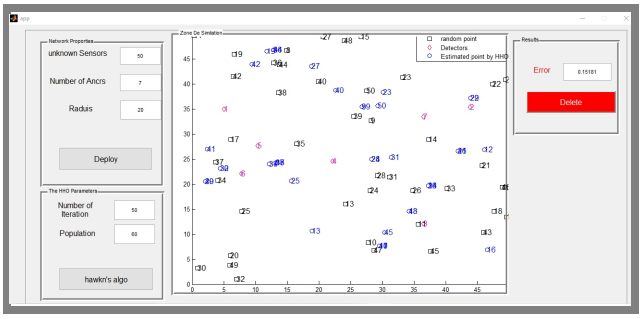

Fig. 2. Example of a localization process relying on the proposed algorithm.

4.1. Effect of Anchor Density

The number of anchor nodes is one of the most important parameters that influences the localization accuracy in WSNs. In this experiment, the objective is to evaluate the impact of

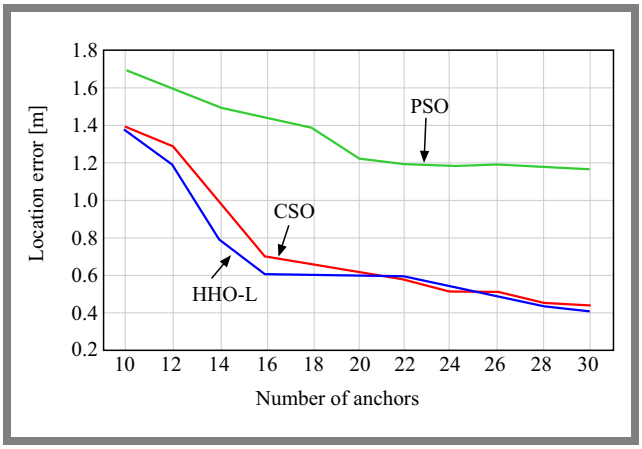

Fig. 3. Anchor effect on localization error.

a varying the number of anchors on localization performance. To achieve this, several experiments were conducted in which the number of anchors varied from 10 to 30. All other network parameters remained unchanged, as detailed in Tab. 1.

Figure 3 presents the localization error obtained as the number of anchors changes, using the HHO algorithm, along with the two other metaheuristic algorithms employed for comparison. This evaluation allows to assess the sensitivity of each algorithm to the density of anchor nodes in the network.

The average localization error was evaluated by varying the number of anchors in the network at different node densities. The results clearly show that as the number of anchors increases from 10 to 30, the localization accuracy improves significantly across all algorithms used in this comparative study. The proposed approach (HHO-based localization) consistently outperforms other methods, demonstrating its effectiveness.

This improvement is attributed to the fact that, as the number of anchor nodes increases, the number of unknown nodes that can estimate their positions based on these reference anchors also increases. Anchors are typically deployed manually or are equipped with GPS receivers, making them reliable reference points.

However, it is observed that beyond a certain threshold, a further increase in the number of anchors yields diminishing returns in terms of localization accuracy. The improvement becomes minimal or negligible, suggesting that an excessive number of anchors may not be effective, especially considering the additional hardware (GPS modules) required or the need to manually place the anchors.

Furthermore, the results also indicate that as the density increases, the average localization error decreases, reinforcing the importance of the network structure in localization performance. Therefore, selecting an optimal anchor-to-node density ratio is essential to balance performance and cost in real-world wireless sensor network deployments.

4.2. Effect of Unknown Sensor Density

Localization performance is also impacted by the density of unknown sensor nodes. In this experiment, we varied the number of sensor nodes, while keeping all other parameters unchanged, as specified in Tab. 1. The results of this experiment are illustrated in Fig. 4.

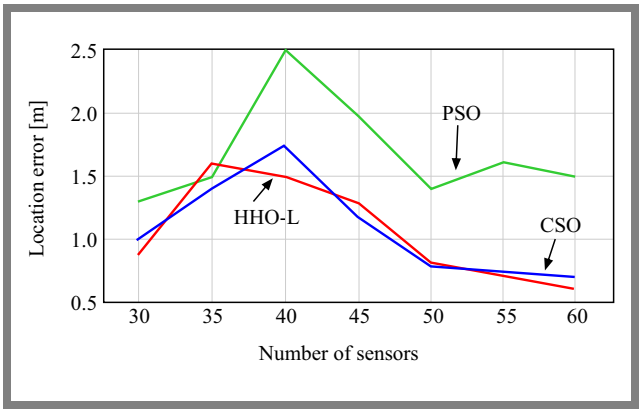

Fig. 4. Effect of unknown sensors on localization error.

One may notice that the localization error decreases gradually as the number of nodes increases. This improvement is attributed to the fact that a higher density of unknown sensors enhances the connectivity of the network. Consequently, the likelihood that each unknown node has multiple anchor nodes within its communication range increases. This means that nodes are more evenly and densely distributed throughout the network, allowing them to form connections with several neighbors and effectively participate in the localization process.

Furthermore, in the proposed approach, once unknown nodes are successfully localized, they can be reused as additional anchor nodes to assist in the localization of other nodes that lack at least three anchors in their vicinity. This dynamic anchor promotion strategy significantly contributes to the superiority of our approach, even with different node densities, as confirmed by experimental results.

4.3. Effect of Connectivity Radius

Connectivity radius is another key parameter that significantly affects the accuracy of node localization in WSNs. The influence of the connectivity radius on the performance of the HHO algorithm for localization is illustrated in Fig. 4. This evaluation verifies localization error, considering different network settings, and includes a comparison with two alternative metaheuristic approaches.

In this experiment, the connectivity radius is varied within the 5 – 20 m range. As shown in Fig. 5, when the connectivity radius is less than 10 m, the internode communication is relatively weak, leading to poor connectivity across the network. As a result, the average localization error is slightly higher, mainly due to the fact that many unknown nodes cannot access a sufficient number of anchor nodes to perform an accurate localization.

However, as the connectivity radius increases above 18 m, the localization error begins to decrease. This improvement is due to the fact that more unknown nodes are able to detect multiple anchors within their communication range, providing more distance-based information to accurately estimate their positions. Nevertheless, the decrease in error becomes more

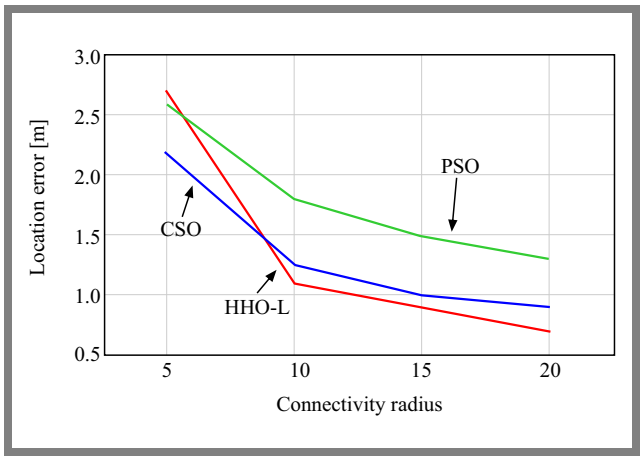

Fig. 5. Effect of connectivity radius on localization error.

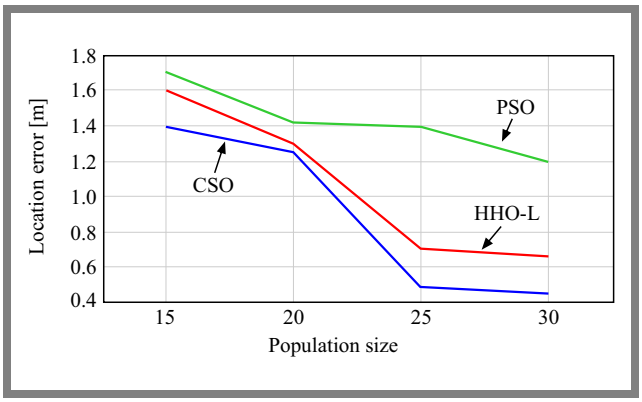

Fig. 6. Effect of population size on localization error.

gradual beyond a certain point, indicating a saturation effect, where further increases in radius yield marginal improvements only.

These results confirm that the selection of an appropriate connectivity radius is essential for maintaining sufficient network connectivity and achieving high localization accuracy while avoiding unnecessary energy consumption or communication overhead.

4.4. Effect of Population Size

The population size, i.e., the number of individuals in the swarm, plays an important role in the convergence behavior of metaheuristic algorithms, particularly during the exploration phase of the search space. In addition, the size of the population directly influences the execution time of the algorithm. Similarly, variations in the number of iterations can have a comparable effect on performance and computational cost.

The relationship between localization error and population size is illustrated in Fig. 6. The results show that when the number of individuals in the population is small, the localization accuracy is relatively low.

However, as the population size increases, accuracy improves as well. This improvement is attributed to the algorithm's ability to better exploit the search space during the exploitation phase. A larger population allows the algorithm to explore the neighborhood of each solution more effectively in search of the optimal coordinates of the unknown nodes.

In essence, a larger population enhances the algorithm's ability to avoid premature convergence and reduces the risk of falling into a local optimum. Therefore, an adequately large number of individuals is necessary to ensure effective localization and to improve the robustness of the solution, especially in complex or high-dimensional problem spaces.

5. Conclusions

The proposed method leverages the dynamic transition between the exploration and exploitation phases inherent to HHO, enabling robust performance under varying deployment conditions. Simulation experiments demonstrated that HHO-L achieves approximately 25% higher localization accuracy compared to other bio-inspired techniques. Specifically, it reached an average accuracy of approximately 0.4 m, outperforming CSO (0.5 m) and significantly surpassing PSO (1.2 m). The study relies on Matlab simulations without any physical testbeds. In real deployments, RSSI fluctuations may be more severe, which could degrade accuracy.

These results not only confirm the algorithm's superiority in classical WSN contexts but also highlight its potential for integration into emerging 6G CSC environments, where precise localization improves semantic compression, spectrum utilization, and adaptive communication in intelligent transportation.

Future research can build on this work in several directions:

- Adaptive and hybrid strategies – dynamically adjusting hawk population size, escape energy parameters, or integrating HHO with complementary metaheuristics (e.g., grey wolf optimizer, whale optimization algorithm) could further improve convergence speed and resilience against environmental noise.
- Cross-layer integration in 6G CSC – embedding HHO-L within semantic-aware network architectures, enabling joint optimization of localization, routing, and spectrum allocation for applications such as V2X, UAV-assisted sensing, and RIS-enabled communication.
- Scalability and real-world deployment – extending the framework to large-scale heterogeneous WSNs with irregular topologies and mobility patterns, and validating its performance on physical testbeds.
- Real hardware experiments, study of environmental noise, energy consumption, and scalability.
- Energy-aware localization – combining HHO-L with energy-efficient communication protocols to minimize localization overhead in resource-constrained next-generation sensor deployments.

By addressing these aspects, HHO-L can evolve from a high-accuracy localization algorithm into a core enabler for intelligent, semantic-driven wireless sensor networks in the era of 6G and beyond.

References

[1] D.W. Wajgi and J.V. Tembhurne, "Localization in Wireless Sensor Networks and Wireless Multimedia Sensor Networks Using Clustering Techniques", *Multimedia Tools and Applications*, vol. 83, pp. 6829–6879, 2023 (https://doi.org/10.1007/s11042-023-15956-z).

[2] M.M. Saeed *et al.*, "A Comprehensive Survey on 6G-security: Physical Connection and Service Layers", *Discover Internet of Things*, vol. 5, art. no. 28, 2025 (https://doi.org/10.1007/s43926-025-00123-7).

[3] L. Doherty, K.S.J. Pister, and L. El Ghaoui, "Convex Position Estimation in Wireless Sensor Networks", *Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies*, Anchorage, USA, 2001 (https://doi.org/10.1109/INFCOM.2001.916662).

[4] K. Saleem *et al.*, "Intelligent Multi-agent Model for Energy-efficient Communication in Wireless Sensor Networks", *EURASIP Journal on Information Security*, art. no. 9, 2024 (https://doi.org/10.1186/s13635-024-00155-6).

[5] Y. Zhao *et al.*, "Real-time Localization in Wireless Sensor Network with Multimedia Applications", *Multimedia Tools and Applications*, vol. 77, pp. 21791–21801, 2018 (https://doi.org/10.1007/s11042-017-5506-z).

[6] S. Yun *et al.*, "A Soft Computing Approach to Localization in Wireless Sensor Networks", *Expert Systems with Applications*, vol. 36, pp. 7552–7561, 2009 (https://doi.org/10.1016/j.eswa.2008.09.064).

[7] A. Gopakumar and L. Jacob, "Localization in Wireless Sensor Networks Using Particle Swarm Optimization", *IET International Conference on Wireless, Mobile and Multimedia Networks*, Beijing, China, 2008.

[8] A. Kumar, A. Khosla, J.S. Saini, and S. Singh, "Computational Intelligence Based Algorithm for Node Localization in Wireless Sensor Networks", *2012 6th IEEE International Conference on Intelligent Systems*, Sofia, Bulgaria, 2012 (https://doi.org/10.1109/IS.2012.6335173).

[9] X.-B. Meng, Y. Liu, X. Gao, and H. Zhang, "A New Bio-inspired Algorithm: Chicken Swarm Optimization", *Advances in Swarm Intelligence*, pp. 86–94, 2014 (https://doi.org/10.1007/978-3-319-11857-4_10).

[10] S. Goyal and M.S. Patterh, "Wireless Sensor Network Localization Based on Cuckoo Search Algorithm", *Wireless Personal Communications*, vol. 79, pp. 223–234, 2024 (https://doi.org/10.1007/s11277-014-1850-8).

[11] S. Rabhi, F. Semcheddine, and N. Mbarek, "An Improved Method for Distributed Localization in WSNs Based on Fruit Fly Optimization Algorithm", *Automatic Control and Computer Sciences*, vol. 55, pp. 286–296, 2021 (https://doi.org/10.3103/S0146411621030081).

[12] M. Miloud, R. Abdellatif, and P. Lorenz, "Moth Flame Optimization Algorithm Range-based for Node Localization Challenge in Decentralized Wireless Sensor Network", *International Journal of Distributed Systems and Technologies (IJDST)*, vol. 10, pp. 82–109, 2019 (https://doi.org/10.4018/IJDST.2019010106).

[13] Z. Lalama, Zahia, S. Boulfekhar, and F. Semechedine, "Localization Optimization in WSNs Using Meta-heuristics Optimization Algorithms: A Survey", *Wireless Personal Communications*, vol. 122, pp. 1197–1220, 2022 (https://doi.org/10.1007/s11277-021-08945-8).

[14] H. Liouane *et al.*, "Regularized Least Square Multi-hops Localization Algorithm for Wireless Sensor Networks", *IEEE Access*, vol. 9, pp. 136406–136418, 2021 (https://doi.org/10.1109/ACCESS.2021.3116767).

[15] S. Messous, H. Liouane, O. Cheikhrouhou, and H. Hamam, "Improved Recursive DV-hop Localization Algorithm with RSSI Measurement for Wireless Sensor Networks", *Sensors*, vol. 21, art. no. 4152, 2021 (https://doi.org/10.3390/s21124152).

[16] S.K. Haider *et al.*, "Energy Efficient UAV Flight Path Model for Cluster Head Selection in Next Generation Wireless Sensor Networks", *Sensors*, vol. 21, art. no. 8445, 2021 (https://doi.org/10.3390/s21248445).

[17] D. Karunanidy *et al.*, "An Intelligent Optimized Route-discovery Model for IoT-Based VANETs", *Processes*, vol. 9, art. no. 2171, 2021 (https://doi.org/10.3390/pr9122171).

[18] A.A. Heidari *et al.*, "Harris Hawks Optimization: Algorithm and Applications", *Future Generation Computer Systems*, vol. 97, pp. 849–872, 2019 (https://doi.org/10.1016/J.FUTURE.2019.02.028).

[19] K. Sheth *et al.*, "A Taxonomy of AI Techniques for 6G Communication Networks", *Computer Communication*, vol. 161, pp. 279–303, 2020 (https://doi.org/10.1016/j.comcom.2020.07.035).

[20] E. Strinati *et al.*, "Goal-oriented and Semantic Communication in 6G AI-native Networks: The 6G-GOALS Approach", *2024 Joint European Conference on Networks and Communications & 6G Summit (EuCNC/6G Summit)*, Antwerp, Belgium, 2024 (https://doi.org/10.1109/EuCNC/6GSummit60053.2024.10597087).

[21] Y. Fu, W. Cheng, W. Zhang, and J. Wang, "Scalable Extraction Based Semantic Communication for 6G Wireless Networks", *IEEE Communications Magazine*, vol. 62, pp. 96–102, 2024 (https://doi.org/10.1109/MCOM.021.2300269).

[22] H. Wei *et al.*, "Localization Performance Analysis Based on Channel Knowledge Map", *Physical Communication*, vol. 72, art. no. 102721, 2025 (https://doi.org/10.1016/j.phycom.2025.102721).

[23] J. Li *et al.*, "Vehicle Positioning Method Based on UAV Assisted RIS Integrated Sensing and Communication System", *Physical Communication*, vol. 72, art. no. 102780, 2025 (https://doi.org/10.1016/j.phycom.2025.102780).

[24] P. Dong, Q. Wu, X. Zhang, and G. Ding, "Edge Semantic Cognitive Intelligence for 6G Networks: Novel Theoretical Models, Enabling Framework, and Typical Applications", *China Communications*, vol. 19, pp. 1–14, 2022 (https://doi.org/10.23919/JCC.2022.08.001).

[25] Q. Cui *et al.*, "Overview of AI and Communication for 6G Network: Fundamentals, Challenges, and Future Research Opportunities", *Science China Information Sciences*, vol. 68, art. no. 171301, 2024 (https://doi.org/10.1007/s11432-024-4337-1).

[26] S. Sharif, F. Khandaker, and W. Ejaz, "Semantic Communication: Implication for Resource Optimization in 6G Networks", *2024 IEEE International Conference on Advanced Telecommunication and Networking Technologies (ATNT)*, Johor Bahru, Malaysia, 2024 (https://doi.org/10.1109/ATNT61688.2024.10719121).

[27] C. Wang *et al.*, "Multimodal Semantic Communication Accelerated Bidirectional Caching for 6G MEC", Future Generation Computer Systems, vol. 140, pp. 225–237, 2022 (https://doi.org/10.1016/j.future.2022.10.036).

[28] M. Chen *et al.*, "Cross-modal Graph Semantic Communication Assisted by Generative AI in the Metaverse for 6G", *Research*, vol. 7, art. no. 0342, 2024 (https://doi.org/10.34133/research.0342).

[29] Y. Sanjalawe *et al.*, "A Review of 6G and AI Convergence: Enhancing Communication Networks with Artificial Intelligence", *IEEE Open Journal of the Communications Society*, vol. 6, pp. 2308–2355, 2025 (https://doi.org/10.1109/OJCOMS.2025.3553302).

[30] A.K. Abasi, M. Aloqaily, M. Guizani, and B. Ouni, "Metaheuristic Algorithms for 6G Wireless Communications: Recent Advances and Applications", *Ad Hoc Networks*, vol. 158, art. no. 103474, 2024 (https://doi.org/10.1016/j.adhoc.2024.103474).

[31] A.K. Abasi *et al.*, "A Survey on Securing 6G Wireless Communications Based Optimization Techniques", *2023 International Wireless Communications and Mobile Computing (IWCMC)*, Marrakesh, Morocco, 2023 (https://doi.org/10.1109/IWCMC58020.2023.10183210).

[32] D.K. Jain *et al.*, "Metaheuristic Optimization-based Resource Allocation Technique for Cybertwin-driven 6G on IoE Environment", *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 4884–4892, 2022 (https://doi.org/10.1109/TII.2021.3138915).

[33] Y. Lil, X. Zhou, and J. Zhao, "Resource Allocation for Semantic Communication Under Physical-layer Security", *GLOBECOM 2023 – 2023 IEEE Global Communications Conference*, Kuala Lumpur, Malaysia, 2023 (https://doi.org/10.1109/GLOBECOM54140.2023.10437297).

[34] W. Yang *et al.*, "Semantic Communication Meets Edge Intelligence", *IEEE Wireless Communications*, vol. 29, pp. 28–35, 2022 (https://doi.org/10.1109/MWC.004.2200050).

[35] S.R. Pokhrel and J. Choi, "Understand-before-talk (UBT): A Semantic Communication Approach to 6G Networks", *IEEE Transactions on Vehicular Technology*, vol. 72, pp. 3544–3556, 2022 (https://doi.org/10.1109/TVT.2022.3219363).

Seddik Rabhi, Ph.D.
Laboratory of Mathematics Modeling and Applications
https://orcid.org/0000-0002-1352-3279
E-mail: rabhi_seddik@yahoo.fr
University of Adrar, Adrar, Algeria
https://en.univ-adrar.edu.dz

Low-complexity Optimized Version of AOR Algorithm for Signal Precoding in Large-scale MIMO Systems

Naceur Aounallah and Smail Labed

Kasdi Merbah Ouargla University, Ouargla, Algeria

https://doi.org/10.26636/jtit.2025.4.2281

Abstract — In recent years, there has been a growing focus on research concerning wireless communication technologies, with a particular emphasis placed on the emerging field of massive MIMO systems. In these systems, precoding performed at the base station (BS) is a crucial signal processing task which ensures reliable downlink transmission. In this paper, we propose a new modified accelerated overrelaxation (AOR) approach to enhance signal precoding in large-scale MIMO downlink systems. This approach uses distinctive matrix decompositions along with optimally selected relaxation and acceleration parameters. Specifically, the proposed method, termed "optimized symmetric accelerated over-relaxation (OSAOR)", exhibits two key advantages: low complexity (compared to the near optimal zero forcing (ZF) precoder) and iterative nature, with its parameters optimized by means of the particle swarm optimization (PSO) algorithm that is capable of boosting convergence and improving precoding precision. Simulation results are given to confirm the superiority of the proposed algorithm, as it may outperform conventional AOR and other existing solutions.

Keywords — AOR iteration, linear precoders, low complexity, massive MIMO, PSO algorithm

1. Introduction

In recent years, research focusing on the field of communications has revealed that the full potential of MIMO systems to achieve high spectral efficiency remains underutilized in practical applications. As a result, the large-scale MIMO (LS-MIMO) approach, commonly referred to as massive MIMO, has emerged as a promising solution to address this limitation. This technology involves equipping base stations with large antenna arrays that contain tens to hundreds of elements, thus enabling simultaneous communication with multiple terminals over the same time-frequency resources.

Using this approach, massive MIMO significantly enhances spectral efficiency, allowing higher data rates without increasing bandwidth. Furthermore, it provides substantial benefits in terms of capacity, energy efficiency, and link reliability, making it a key enabler for 5G and beyond wireless networks [1]. As demand for high-speed, low-latency and ultra-reliable communication continues to grow, massive MIMO stands out as a transformative wireless access technology poised to meet the evolving needs of next-generation communication systems [2], [3].

In addition to the promising advantages, several practical challenging problems must be solved to realize a massive MIMO system, with precoding performed in the system's downlink portion being one of them. Consequently, a plethora of massive MIMO precoding methods has been proposed to reach a satisfactory trade-off between overall performance and the complexity of the solution.

In fact, precoding methods can be categorized as linear or non-linear, depending on whether they involve non-linear operations in their computation. Non-linear precoding techniques, such as constant envelope (CE), dirty paper coding (DPC) [4], vector perturbation (VP), lattice-aided methods, and Tomlinson-Harashima precoding (THP) are generally impractical for hardware implementation due to their high computational complexity [5]. As a result, linear precoding techniques, including matched filter (MF), zero-forcing (ZF), and minimum mean square error (MMSE) are preferred and considered as benchmark linear precoders. However, these methods require that the channel matrix, including all user data, be inversed [6].

To address high computational complexity associated with large-scale matrix inversion, several alternative linear precoding methods have been proposed. These methods generally fall into three main categories: direct methods, iterative methods, and expansion methods. Direct methods transform the matrix requiring inversion into a product of simpler matrices, such as QR or Cholesky decomposition [7]. Although accurate, these methods suffer from high computational complexity.

Iterative methods solve linear equations through successive approximations and include techniques such as the Richardson method [8], the conjugate gradient (CG) method [9], and successive over-relaxation (SOR) [10]. These approaches are computationally efficient for large-scale systems, but may require careful parameter tuning for fast convergence.

Expansion methods approximate the matrix inverse using a series of matrix vector products, with examples including the Neumann series (NS) [11], the truncated polynomial expansion (TPE), as well as Newton iteration (NI) and Chebyshev iteration (CI) algorithms [12], [13]. These approaches provide a trade-off between complexity and accuracy, making them particularly suitable for massive MIMO systems.

Conventional quasi-optimum precoders, such as zero forcing (ZF) and minimum mean square error (MMSE) [14], achieve the best BER performance. However, they require matrix inversion calculation, which leads to high computational complexity.

To address this challenge, researchers have explored iterative algorithms to overcome the need for direct matrix inversion in mathematical operations [15]. The accelerated overrelaxation (AOR) method, introduced in [16], is one of such approaches. It bypasses the operation of the inverse matrix through linear iteration, thus reducing computational complexity from $\mathcal{O}(U^3)$ to $\mathcal{O}(U^2)$. Despite this improvement, AOR performance remains suboptimal and requires further enhancement. In response, the authors of [17] proposed the symmetric accelerated over-relaxation (SAOR) method, which builds upon AOR by incorporating two similar symmetric matrices for iteration, leading to improved performance over the original AOR method.

Building on these developments, recent research has focused on further improvement of iterative AOR-based precoding schemes. For example, in [18], an improved AOR-based precoding method was introduced, where the acceleration factor of AOR is optimized using the relationship between the spectral radius and the eigenvalues of a positive definite Hermitian matrix, which depends solely on the number of transmitting and receiving antennas.

In addition, article [19] presents an improved variant of the traditional AOR method, termed optimized AOR (OAOR). This approach integrates a novel variant of meta-heuristic particle swarm optimization (PSO), refining the cognitive coefficients to optimize the relaxation parameters for OAOR. Furthermore, the AOR iterative method, which is applicable both to massive MIMO detection and precoding, has been further refined in [20] using the Nelder-Mead simplex optimization technique. This heuristic optimization approach facilitates the determination of optimal acceleration and relaxation parameters, leading to higher detection accuracy and faster convergence.

The main contribution of this work is the proposition of a new version of the accelerated over-relaxation (AOR) algorithm that is used as a linear precoder for massive MIMO systems. The new solution is referred to as the optimized symmetric accelerated over-relaxation (OSAOR) algorithm. The proposed scheme is developed by integrating the strengths of the symmetric accelerated overrelaxation (SAOR) method and the PSO algorithm.

The primary objective of employing PSO is to find the optimal values of relaxation and acceleration parameters for the SAOR precoder. We analyze the computational complexities of SAOR and OSAOR precoders and compare them with those of the near-optimal ZF precoder. Additionally, we discuss the influence of the relaxation and acceleration parameters on the convergence speed of the proposed OSAOR method.

The remainder of the paper is organized as follows. In Section 2, we describe the multi-user large MIMO downlink system model, formulate the precoding problem, and review the ZF benchmark precoder. Then, in Section 3, the classical AOR detection algorithm is reviewed and its new version is proposed. Computational complexity is studied in Section 4, and the numerical results allowing to compare the performance of the proposed approach are analyzed in Section 5. The latter is followed by conclusions and future perspectives presented in Section 6.

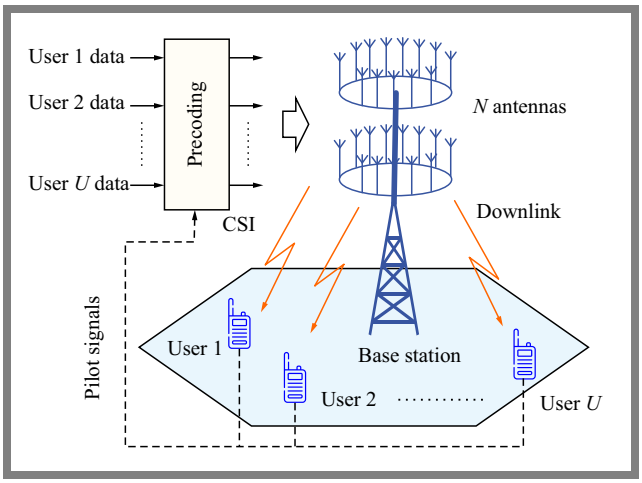

Fig. 1. Model of a massive MIMO downlink system with U single antenna users and N BS antennas.

2. System Description and Problem Formulation

This section examines the downlink transmission of a large-scale multi-user MIMO system, where each base station (BS) is fitted with N antennas and serves concurrently U single-antenna users within the same frequency band, with $U \ll N$ [1]. We assume that the base station has perfect knowledge of channel state information (CSI), which can be obtained through pilot-based training [21]. For this communication scenario, we consider the Rayleigh flat-fading channel model. A model of the system is shown in Fig. 1.

The received signal vector $y \in C^{U\times 1}$ at the user terminals can be mathematically described as follows:

$$y = \sqrt{\rho_r}\, H\, x + n\,, \tag{1}$$

where $\sqrt{\rho_r}$ is a normalization factor to determine signal to noise power ratio (SNR), $H \in C^{U\times N}$ is the channel matrix, $x \in C^{N\times 1}$ is the transmitted symbol vector, and $n \in C^{U\times 1}$ is the complex additive white Gaussian noise (AWGN) vector at the receiver with zero mean and σ^2 variance.

The considered system model can be represented in a matrix form as:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_U \end{bmatrix} = \sqrt{\rho_r} \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1N} \\ h_{21} & h_{22} & \dots & h_{2N} \\ h_{31} & h_{32} & \dots & h_{3N} \\ \vdots & \vdots & \ddots & \vdots \\ h_{U1} & h_{U2} & \dots & h_{UN} \end{bmatrix}^T \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ \vdots \\ n_U \end{bmatrix} . \tag{2}$$

Each entry h_{ij} of channel matrix H is modeled as an independent and identically distributed (i.i.d.) complex Gaussian random variable according to $\mathcal{CN}(0,1)$ distribution, while the AWGN vector entries are i.i.d., with each of them following $\mathcal{CN}(0,\sigma^2)$ distribution.

In practice and under nonideal conditions, the assumption of perfect CSI at the BS is rarely achievable due to noisy channel estimation and pilot contamination. To model this, the estimated channel matrix $\hat{H}$ can be represented as [22]:

$$\hat{H} = \sqrt{1-\tau^2}\, H + \tau H_e \,, \tag{3}$$

where H denotes the true channel matrix with i.i.d. entries following $\mathcal{CN}(0,1)$, $H_e \in C^{U\times N}$ being the estimation error matrix with i.i.d. entries also distributed as $\mathcal{CN}(0,1)$, and $0 \leqslant \tau \leqslant 1$ controlling the quality of CSI.

The challenge of signal precoding in massive MIMO at the base station is to optimize the transmitted signal vector x, so that users receive their intended signal via the received vector y with minimal interference.

Therefore, the base station applies a precoding matrix $W \in C^{N\times U}$ to the original data symbols $s \in C^{U\times 1}$ intended for users, leading to:

$$x = Ws \,. \tag{4}$$

Thus, the received signal becomes:

$$y = \sqrt{\rho_r}\, HWs + n = \sqrt{\rho_r}\, Q\, s + n \,, \tag{5}$$

where $Q = HW$ represents the equivalent channel matrix, and making it close to an identity matrix helps minimize multi-user interference.

The signal-to-interference plus noise ratio (SINR) at the u-th reception can be given to the user as follows [23]:

$$SINR_u = \frac{\frac{\rho_r}{U}\,|q_{u,u}|^2}{\frac{\rho_r}{U}\sum\limits_{m\neq n}^{U}|q_{m,u}|^2 + 1} = \frac{\rho_r}{U}\,|q_{u,u}|^2 \,, \tag{6}$$

where $q_{m,u}$ denotes the element of matrix Q at the intersection of the m-th row and the u-th column.

The ergodic capacity of the downlink massive MIMO system after applying precoding can be expressed as [22]:

$$C = \sum_{u=1}^{U} \log_2(SINR_u + 1) \,. \tag{7}$$

The achievable capacity is a key factor in assessing the performance of precoding techniques.

In this study, the basic zero forcing (ZF) precoding algorithm is used as a benchmark linear precoder. Its primary objective is to mitigate interuser interference by adhering to optimization criteria designed to minimize it [24]. The corresponding ZF precoding matrix is the following:

$$W_{ZF} = \beta\, H^H (HH^H)^{-1} = \beta\, H^H G^{-1} \,, \tag{8}$$

where $G = HH^H$ stands for the Gram matrix, while β is the normalization parameter that accounts for the average fluctuations in transmit power. This parameter is defined as:

$$\beta = \sqrt{\frac{U}{tr(G^{-1})}} \,, \tag{9}$$

The transmitted signal, after ZF precoding has been applied, can be represented as:

$$x_{ZF} = W_{ZF} s = \beta\, H^H G^{-1} s = \beta\, H^H z \,, \tag{10}$$

where $G^{-1}s = z$, which clearly results in:

$$G\, z = s \,. \tag{11}$$

The vector of the received signals after applying ZF precoding is given by:

$$y_{ZF} = \beta\sqrt{\rho_r} HH^H (HH^H)^{-1} s + n = \beta\sqrt{\rho_r} Es + n \,, \tag{12}$$

where $E = HH^H(HH^H)^{-1}$ is the ZF equivalent channel matrix. Then, the corresponding signal-to-interference-plus-noise ratio (SINR) for any u-th user can be determined as [22], [23]:

$$\begin{aligned} SINR_u &= \frac{\frac{\rho_r}{U}|e_{u,u}|^2}{\frac{\rho_r}{U}\sum\limits_{m\neq u}^{U}|e_{m,u}|^2 + 1} \\ &= \frac{\rho_r}{U}\left|e_{u,u}\right|^2 = \frac{\rho_r}{tr(G^{-1})} \,, \end{aligned} \tag{13}$$

where $e_{m,k}$ represents the element of matrix E located in the m-th row and the k-th column.

Based on Eq. (13), the sum capacity achieved by ZF precoding in a large-scale MIMO system can be determined using the following expression [23], [25]:

$$\begin{aligned} C_{ZF} &= \sum_{u=1}^{U} \log_2(SINR_u + 1) \\ &= U \log_2\left(\frac{\rho_r}{tr(G^{-1})} + 1\right) \,. \end{aligned} \tag{14}$$

3. Precoding Algorithms

3.1. Standard AOR

The standard accelerated over-relaxation (AOR) splitting method was originally proposed in [16] as a two parameter linear stationary method for solving linear systems of the $x = A^{-1}\, b$ form. Hadjidimos demonstrated that when these two parameters are appropriately chosen, the AOR method achieves faster convergence compared to other similar approaches.

The AOR method has been proven to be a powerful tool for solving problems associated with linear systems [18]. To explicitly formulate the iterative process of the AOR method, we start by decomposing the precoding matrix W from Eq. (4) into its fundamental components as follows:

$$W = W_D - W_L - W_U \,, \tag{15}$$

where W_D represents the diagonal part of W, while $-W_L$ and $-W_U$ correspond to the strictly lower and strictly upper triangular parts, respectively.

Using this decomposition, we can now reformulate the original Eq. (4) as follows:

$$(W_D - W_L - W_U)\, x = s \,. \tag{16}$$

Rewrite this equation:

$$W_D\, x = W_L\, x + W_U\, x + s\,. \tag{17}$$

We apply W_D^{-1}, the inverse of the diagonal matrix W_D, to both sides and simplify the left side, since $W_D^{-1}\, W_D = I$:

$$x = W_D^{-1}(W_L\, x + W_U\, x + s)\,. \tag{18}$$

At this stage, we incorporate the relaxation parameter $\omega \in R$:

$$x = x + \omega\left(W_D^{-1}(W_L\, x + W_U\, x + s) - x\right). \tag{19}$$

Expanding the equation, the following form is obtained:

$$x = x + \omega\,(W_D^{-1}\, W_L\, x + W_D^{-1} W_U\, x + W_D^{-1} s - x)\,. \tag{20}$$

Next, we introduce the acceleration parameter $\gamma \in R$:

$$x = x + \omega\,(\gamma\, W_D^{-1} W_L\, x + W_D^{-1} W_U\, x + W_D^{-1} s - x)\,. \tag{21}$$

Reorganizing the terms:

$$x = (1-\omega)\, x + \omega\,\gamma\, W_D^{-1} W_L\, x + \omega\, W_D^{-1} W_U\, x + \omega\, W_D^{-1} s\,. \tag{22}$$

We extract x as a common factor:

$$x = \left[(1-\omega) I + \omega\,\gamma\, W_D^{-1} W_L + \omega\, W_D^{-1} W_U\right] x + \omega\, W_D^{-1} s\,. \tag{23}$$

To formulate this as an iterative method, we introduce the iteration index i:

$$x^{(i+1)} = \left[(1-\omega) I + \omega\,\gamma\, W_D^{-1} W_L + \omega\, W_D^{-1} W_U\right] x^{(i)} + \omega\, W_D^{-1} s\,. \tag{24}$$

To simplify the computation, we multiply both sides by $(W_D - \gamma W_L)$:

$$\begin{aligned}(W_D - \gamma W_L)\, x^{(i+1)} = \big[&(1-\omega) W_D + (\gamma - \omega) W_L \\ &+ \omega\, W_U\big] x^{(i)} + \omega\, s\,.\end{aligned} \tag{25}$$

Finally, solving for $x^{(i+1)}$, we derive the AOR iteration formula:

$$\begin{aligned}x^{(i+1)} = (W_D - \gamma\, W_L)^{-1}\big(\big[&(1-\omega) W_D \\ &+ (\gamma - \omega) W_L + \omega\, W_U\big] x^{(i)} + \omega\, s\big)\,.\end{aligned} \tag{26}$$

The optimal relaxation parameter ω and the optimal acceleration parameter γ of the AOR iterative method are given by [26] as follows:

$$\omega = \frac{2}{1 + \sqrt{1 - \mu_{max}^2}}\,, \tag{27}$$

$$\gamma = \frac{\mu_{max}^2 - \mu_{min}^4}{\mu_{max}^2 (1 - \mu_{min}^2)}\,, \tag{28}$$

where μ_{max} and μ_{min} denote the maximum and the minimum eigenvalues of the Jacobi matrix, respectively.

3.2. Symmetric AOR Version

The symmetric accelerated over-relaxation (SAOR) method can be interpreted as a double sweep approach. The first sweep follows the standard AOR method, while the second sweep applies the AOR method with the roles of W_L and W_U matrices interchanged.

In other words, each iteration of the SAOR method consists of two half-iterations: a forward pass using the AOR method, followed by a backward pass where the equations are processed in a reverse order, effectively applying AOR again.

Therefore, the SAOR iterative process is executed in the two steps described below:

1) Perform the first half iteration, which is equivalent to the AOR iteration [16], as follows:

$$\begin{aligned}x^{(i+1/2)} = (W_D - \gamma W_L)^{-1}\big(\big[&(1-\omega) W_D \\ &+ (\gamma - \omega) W_L + \omega\, W_U\big] x^{(i)} + \omega\, s\big)\end{aligned}\,. \tag{29}$$

2) Perform the second half iteration, applying the AOR method with the equations processed in a reverse order, as follows:

$$\begin{aligned}x^{(i+1)} = (W_D - \gamma\, W_U)^{-1}\big(\big[&(1-\omega) W_D \\ &+ (\gamma - \omega)\, W_U + \omega\, W_L\big]\, x^{(i+1/2)} + \omega\, s\big)\,.\end{aligned} \tag{30}$$

In SAOR-based precoding, both relaxation ω and acceleration γ parameters are within the range of $[0, 2]$, serving the same role as in AOR. Their optimal values can be determined using Eqs. (27) and (28).

3.3. Improved Version of SOAR

The performance of the SAOR version is strongly influenced by its acceleration and relaxation parameters. To systematically determine these parameters, we integrate an innovative approach based on the PSO algorithm. This heuristic method enables the identification of optimal values, ensuring enhanced precoding accuracy and faster convergence, even in complex system configurations.

So, to determine the optimal ω and γ for improving the performance of the SAOR method, we formulate an optimization problem as follows:

$$\begin{aligned}&\text{minimize} \quad f(\omega, \gamma) = \mathbb{E}\left[||x - \hat{x}(\omega, \gamma)||^2\right] \\ &\text{subject to} \quad 0 < \omega < 2, \quad 0 < \gamma < 2 \\ &\rho\left(I - \omega\, W_D^{-1} G^H G + \omega\,\gamma\, {W_D}^{-1}(W_D - W_L - W_U)\right) < 1\end{aligned} \tag{31}$$

where $\rho(\cdot)$ denotes the spectral radius of a matrix.

This optimization problem is formulated to minimize the mean square error while guaranteeing the convergence of the method that maximizes system performance [27]. We employ the PSO algorithm to solve this problem and determine the optimal ω and γ. PSO is a powerful and popular metaheuristic optimization algorithm that simulates the intelligent behavior of a swarm to navigate a complex search space and find optimal solutions.

Mapping the problem to PSO [28]:

- Particle. Each particle in the swarm represents a candidate solution.
- Particle position. It is a vector of the parameters we want to optimize. In this case $\text{position\_X} = [\omega_X, \gamma_X]$.
- Search space. We need to define the valid range for ω and γ and these parameters must be within the range of $[0, 2]$ for convergence.
- Fitness function. This is the most critical part. The fitness function evaluates how good a particle's position $[\omega, \gamma]$ is.

The flow chart of the basic PSO is depicted in Fig. 2.

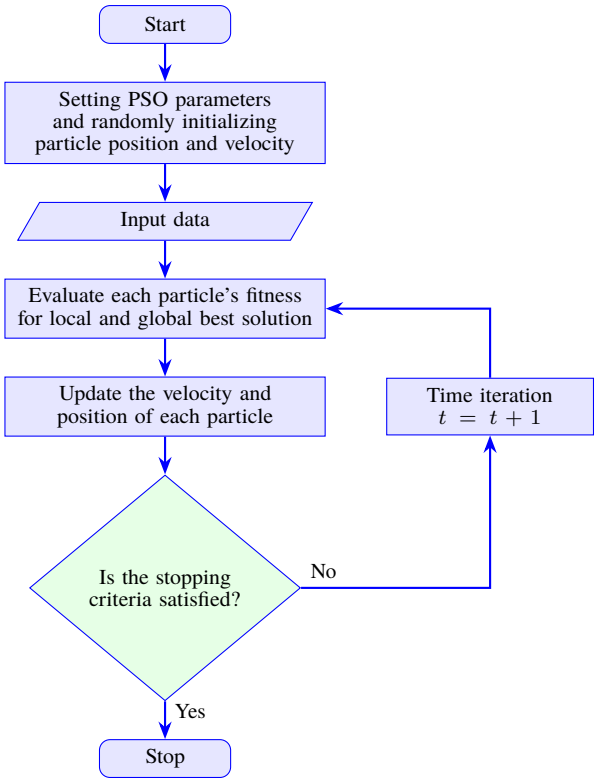

Fig. 2. Particle swarm optimization flow chart.

At each iteration $t+1$, every particle p updates its velocity V and position X. The process is defined by the following velocity and position update equations [29]:

$$V^{(t+1)} = \Omega\, V^{(t)} + c_1\, r_1 \left(p_{best}^{(t)} - X^{(t)}\right) + c_2\, r_2 \left(g_{best}^{(t)} - X^{(t)}\right), \tag{32}$$

$$X^{(t+1)} = X^{(t)} + V^{(t+1)}, \tag{33}$$

where Ω is the inertia weight, c_1 and c_2 are the cognitive acceleration coefficients and r_1, r_2 are random numbers uniformly distributed over the interval $[0,1]^D$; D denotes the dimensionality of the search space (i.e., the problem size).

Unlike OAOR, which optimizes parameters only within the standard AOR framework, the proposed OSAOR combines the symmetric SAOR structure with PSO-based parameter selection.

Since symmetric schemes already improve stability and performance over standard AOR, integrating the PSO-based global optimizer at this level enables further gains in convergence speed and SER performance, making OSAOR fundamentally different from the OAOR precoding algorithm.

4. Computational Complexity

Computational complexity of the SAOR-based precoding algorithm is analyzed in terms of the number of complex multiplications required per iteration. The iterative process is dominated by sparse matrix-vector products and forward/backward substitutions. A full symmetric SAOR iteration consists of one forward and one backward pass. In each pass, the update involves calculating:

$$(\gamma - \omega)\, W_L\, x + (1-\omega)\, W_D\, x + \omega\, W_U\, x\,,$$

which requires approximately

$$\frac{U^2 - U}{2} + U + \frac{U^2 - U}{2} = U^2$$

multiplications, where U is the number of users.

This is followed by a forward substitution to solve the system $(W_D - W_L)\, x_{new}$, which requires an additional $\frac{1}{2}(U^2 - U)$ multiplications.

The sum of operations performed during both passes yields a total complexity of approximately $3\,U^2$ complex multiplications per iteration. Therefore, the SAOR algorithm exhibits a computational complexity of $\mathcal{O}(U^2)$. This is a significant advantage over direct matrix inversion methods such as ZF, which are $\mathcal{O}(U^3)$, making SAOR a highly efficient solution for massive MIMO systems. Although its complexity is slightly higher than $\mathcal{O}(2\,U^2)$ of the SSOR method due to an additional matrix vector product, this cost enables the use of a second parameter γ that can accelerate convergence and improve performance, representing a favorable trade-off between computational effort and system accuracy.

The OSAOR-based precoding algorithm combines SAOR iteration with a PSO stage to optimally tune its relaxation and acceleration parameters. The PSO algorithm operates with P particles over T iterations, with fitness evaluation being the most demanding step. A straightforward evaluation involves operations comparable to a SAOR algorithm with complexity of $\mathcal{O}(U^2)$. However, by adopting sampling and approximation, this can be reduced to $\mathcal{O}(U\log(U))$.

Therefore, the overall cost of the PSO phase is $\mathcal{O}(PTU\log(U))$, which is a one-time cost for a given channel realization and does not contribute to each transmission iteration. After optimization, OSAOR precoding proceeds in a manner that is identical to that charcteristic of SAOR, with per-iteration complexity of $\mathcal{O}(U^2)$. Consequently, the overall complexity of OSAOR remains of order $\mathcal{O}(U^2)$, significantly lower than the complexity of ZF precoding, while providing faster convergence and improved error rate performance.

The computational complexity of linear precoding algorithms is summarized in Tab. 1.

5. Results and Discussion

In this section, we numerically analyze and discuss the performance of the proposed precoding method in terms of SER and ergodic capacity with respect to different SNR values and iteration counts. The results are compared with conventional AOR, SSOR, and SAOR-based multi-user precoders over the Rayleigh fading channel.

For reference, the near optimal ZF precoder is also included as a benchmark. Simulations are carried out for a 128×16 BS user antenna configuration in a massive downlink MIMO system using 64-quadrature amplitude modulation (64-QAM). SER results are averaged over 5×10^4 Monte

Tab. 1. Computational complexity of different precoding schemes.

Precoder	Complexity order	Remarks
ZF	$\mathcal{O}(U^3)$	Requires exact matrix inversion
AOR	$\mathcal{O}(U^2)$	Two parameters (relaxation, acceleration)
SSOR	$\mathcal{O}(U^2)$	Single relaxation parameter
SAOR	$\mathcal{O}(U^2)$	Two parameters (relaxation, acceleration)
OSAOR	$\mathcal{O}(U^2)$	PSO overhead: $\mathcal{O}(PTU \log U)$ (one-time), then $\mathcal{O}(U^2)$ per iteration

Tab. 2. PSO parameters used in the simulation.

Parameter	Value
Number of particles P	20
Number of iterations T	15
Inertia weight (max) $\Omega_{\max}$	0.9
Inertia weight (min) $\Omega_{\min}$	0.4
Cognitive coefficient c_1	2.0
Social coefficient c_2	2.0

Carlo trials, while PSO optimization parameters are listed in Tab. 2.

PSO parameters were selected based on values that are widely adopted in the literature and were validated through preliminary simulations. The size of the swarm $P = 20$ and the iteration limit $T = 15$ provide a good trade-off between reliability of convergence and computational cost. The inertia weight was linearly decreased from $\Omega_{max} = 0.9$ to $\Omega_{min} = 0.4$ to balance exploration and exploitation, while the cognitive and social coefficients were set at $c_1 = c_2 = 2.0$, which are canonical choices that ensure stable convergence. Sensitivity tests confirmed that small variations of these parameters do not significantly affect the convergence behavior or final performance, demonstrating the robustness of the adopted setting.

Figure 3 shows the convergence behavior of the PSO algorithm used to find the optimal parameters for the OSAOR precoder. The fitness function which is defined as $\frac{1}{1+error}$ is designed so that as the error of the precoder decreases towards zero, the fitness value approaches its maximum of 1.

The plot shows that the PSO algorithm converges quickly. There is a significant improvement in fitness between the first and second iterations, indicating that the algorithm rapidly identifies a promising region in the parameter space. From the second iteration onward, the fitness value continues to improve in very fine increments, stabilizing, and effectively converging by the 11-th iteration. This rapid convergence highlights the efficiency of using PSO to tune the parameters of OSAOR.

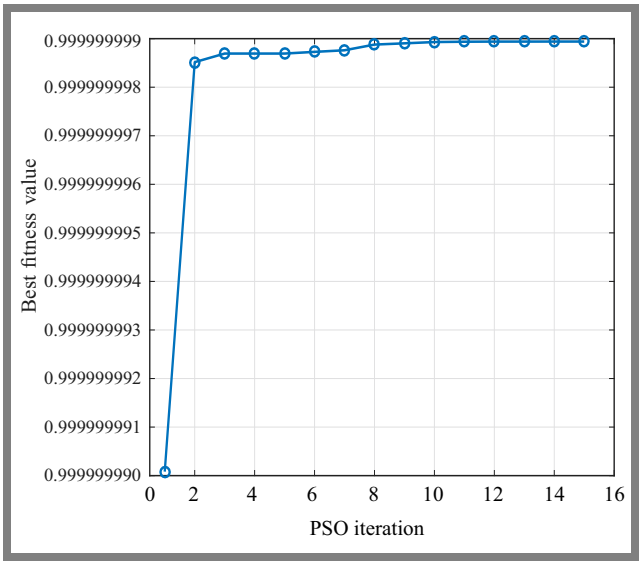

Fig. 3. Convergence of PSO fitness function for optimization of OSAOR parameters.

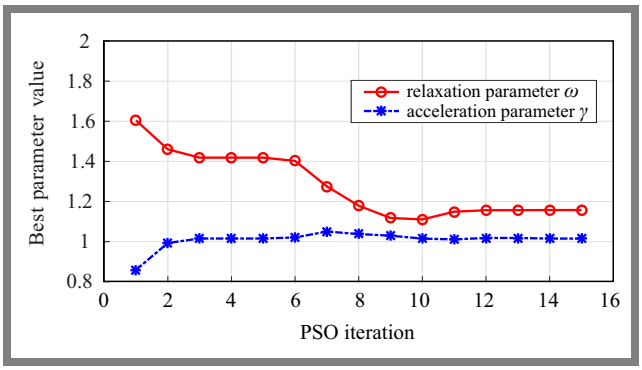

Fig. 4. Evolution of optimal OSAOR parameters ω and γ across PSO iterations.

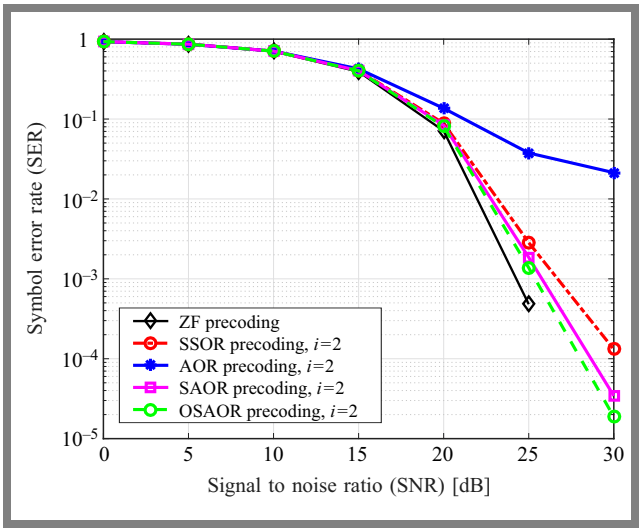

Fig. 5. Comparison of SER vs. SNR performance for a $(N, U) = (128, 16)$ precoding m-MIMO system.

Figure 4 provides the evolution of the two key OSAOR parameters that are being optimized: relaxation parameter ω and acceleration parameter γ. The graph shows that in the initial iterations, particularly for the relaxation parameter ω,

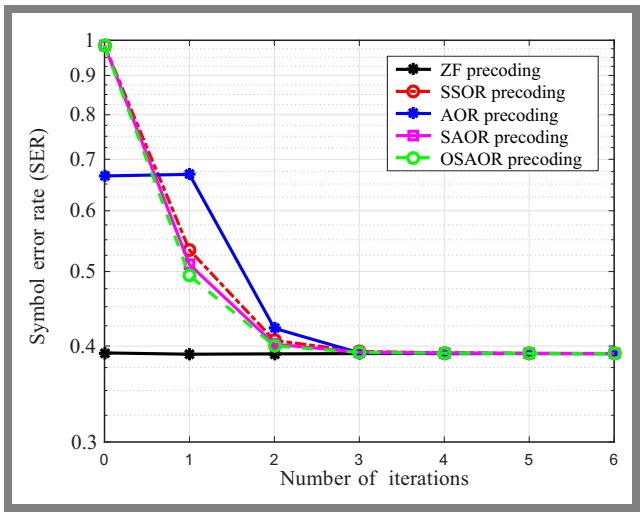

Fig. 6. Comparison of the performance of SER against the number of iterations for a $(N, U) = (128, 16)$ precoding m-MIMO system.

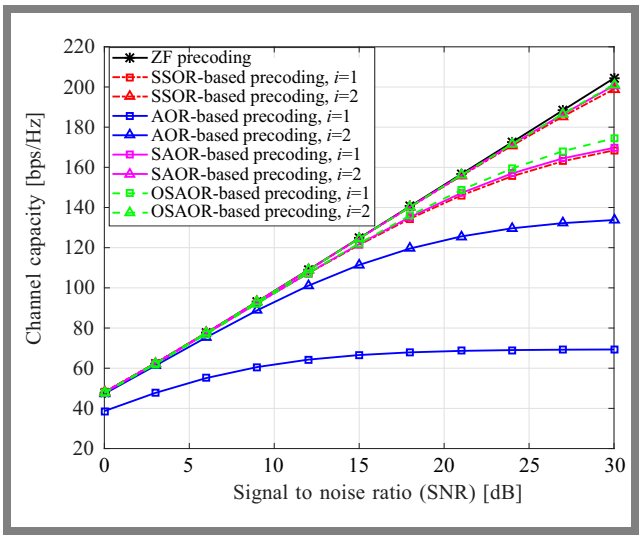

Fig. 7. Comparison of capacity performance of the investigated precoding iterative techniques against the SNR for a $(N, U) = (128, 16)$ m-MIMO system.

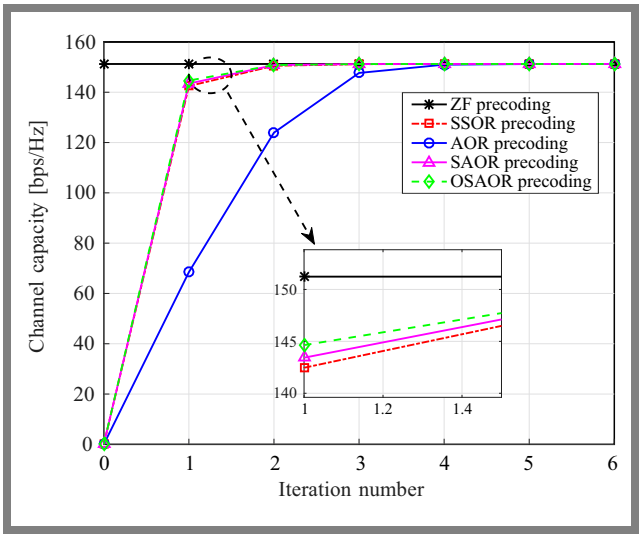

Fig. 8. Comparison of capacity performance of the studied precoding iterative techniques against the number of iterations for a $(N, U) = (128, 16)$ m-MIMO system.

the algorithm makes significant adjustments, starting from approximately 1.6 and decreasing to around 1.1 by iteration 10.

The acceleration parameter γ shows a more modest adjustment, starting near 0.85 and settling around 1.02. After approximately the 10-th iteration, both parameters stabilize, which corresponds directly to the point where the fitness function flattens. This indicates that the PSO algorithm has successfully converged to a stable set of optimal parameters, which are found to be approximately $\omega \simeq 1.15$ and $\gamma \simeq 1.02$ for the given system configuration.

Figure 5 presents the SER performance of the proposed precoding schemes compared to conventional methods for a massive MIMO downlink system with $(N, U) = (128, 16)$ and 64-QAM. As expected, SER decreases as SNR increases for all schemes. The ZF precoder, included as a benchmark, achieves the best performance but at the expense of high computational complexity due to matrix inversion calculations. Among the iterative methods, AOR exhibits the weakest performance, particularly at moderate to high SNR values.

On the contrary, both SSOR and SAOR show significant improvements, with SAOR consistently outperforming SSOR as a result of its acceleration mechanism. In particular, the proposed OSAOR scheme achieves the closest performance to ZF across the entire SNR range, particularly at high SNR, where its SER nearly overlaps with that of ZF. This demonstrates that optimizing SAOR parameters using PSO effectively accelerates convergence and minimizes precoding errors, offering a practical low-complexity alternative to ZF for large-scale MIMO systems.

Figure 6 illustrates the performance of SER versus the number of iterations at SNR = 15 dB. As observed, all iterative precoding methods (SSOR, AOR, SAOR, and OSAOR) start with higher SER but converge rapidly within a few iterations. Notably, OSAOR and SAOR achieve faster convergence, reaching near-optimal performance by the second iteration, whereas SSOR lags in the initial iterations. After approximately three iterations, all methods converge to the same performance level as ZF, demonstrating the efficiency of the proposed OSAOR in reducing complexity while maintaining accuracy.

Channel capacity performance is presented in Fig. 7 as a function of SNR for a $(N, U) = (128, 16)$ precoding massive MIMO system. As expected, the near-optimal ZF precoder achieves the highest capacity across the entire SNR range. Among the iterative schemes, OSAOR- and SAOR-based precoding closely approach the performance of ZF, especially from moderate to high SNR values, while requiring only a small number of iterations.

On the contrary, AOR- and SSOR-based methods exhibit slower capacity growth, with SSOR showing the largest performance gap when compared to ZF. These results confirm that OSAOR provides a favorable trade-off between complexity and spectral efficiency, achieving near-optimal capacity with reduced computational cost.

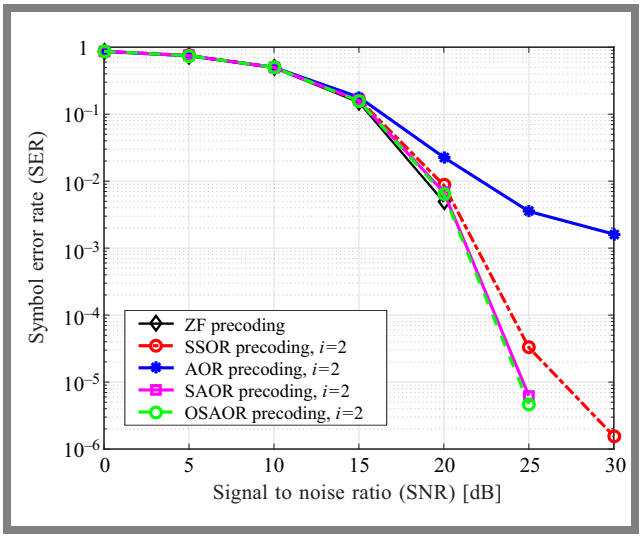

Fig. 9. Comparison of SER performance of the studied precoding iterative techniques against the SNR for a $(N, U) = (256, 32)$ m-MIMO system.

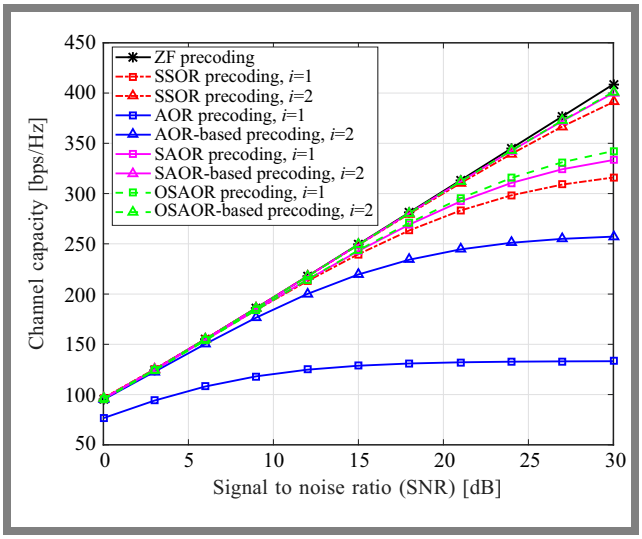

Fig. 10. Comparison of capacity performance of the of the investigated precoding iterative techniques against the SNR for a $(N, U) = (256, 32)$ m-MIMO system.

The convergence performance of several iterative precoding algorithms is illustrated in Fig. 8 by plotting channel capacity against the number of iterations for a massive MIMO system configured with $(N, U) = (128, 16)$ at an SNR of 15 dB. The ZF precoding algorithm serves as a performance benchmark, achieving an optimal channel capacity of approximately 151 bps/Hz. The results clearly demonstrate the superior convergence speed of the symetric-based algorithms (SSOR, SAOR, and OSAOR) over the traditional AOR method. Although the AOR algorithm requires three full iterations to converge to the performance level of ZF, the SSOR, SAOR, and OSAOR schemes achieve near-optimal capacity after just a single iteration and fully converge by the second.

The magnified inset provides a crucial distinction in their first-iteration performance, revealing that the OSAOR algorithm attains the highest capacity, followed closely by SAOR and then SSOR. This signifies that OSAOR offers the most favorable trade-off between computational complexity and performance as it is capable of delivering near-maximum channel capacity with the minimal computational load of a single iteration.

To further evaluate the scalability of the proposed OSAOR algorithm, we extended the simulations to a larger user antenna configuration with $(N, U) = (256, 32)$ using the 32-QAM modulation scheme. Figures 9 and 10 illustrate the resulting SER vs. SNR and achievable capacity vs. SNR, respectively. The results confirm that OSAOR consistently outperforms conventional methods in terms of both error performance and capacity, even under increased system dimensions and other order modulation, thereby demonstrating its robustness and scalability.

6. Conclusions and Future Outlook

By integrating the strengths of the symmetric accelerated over-relaxation (SAOR) method and advantages of the particle swarm optimization (PSO) algorithm, the proposed approach achieves an optimal balance between computational efficiency and SER performance. In addition, the improved version of the standard AOR proposed in this paper achieves a more refined trade-off between performance and complexity compared to basic existing methods.

Furthermore, the simulation results confirm that the proposed OSAOR scheme not only reduces computational load, but also adapts effectively to different system configurations, making it suitable for large-scale deployments. The incorporation of PSO ensures that the relaxation and acceleration parameters are tuned optimally, enabling the algorithm to converge faster and achieve improved error performance.

As this study is limited to simulation-based analysis, future research will extend the scope of the evaluation to cover imperfect CSI scenarios in order to assess the robustness of the proposed method under more realistic channel conditions. Additional performance metrics such as latency, energy efficiency, and computational resource consumption will also be investigated to provide a more comprehensive view of practical applicability.

Furthermore, future work will include hardware-in-the-loop validation and prototype implementations on FPGA/ASIC platforms to evaluate the feasibility of real-time deployment in 5G and beyond wireless systems. Finally, real-time implementation on commercial SDR platforms, followed by field trials, is planned to further confirm the practicality and robustness of the proposed OSAOR scheme under real-world deployment conditions.

Acknowledgments

The work has been funded under the PRFU research project no. A25N01UN300120220001 of the Algerian Ministry of Higher Education and Scientific Research (MESRS).

References

[1] S. Labed and N. Aounallah, "Efficient Iterative Detection Based on Conjugate Gradient and Successive Over-relaxation Methods for Uplink Massive MIMO Systems", *Journal of Telecommunications and Information Technology*, vol. 92, pp. 1–9, 2023 (https://doi.org/10.26636/jtit.2023.169023).

[2] T.A. Sheikh, J. Bora, and M.A. Hussain, "Combined User and Antenna Selection in Massive MIMO Using Precoding Technique", *International Journal of Sensors, Wireless Communications and Control*, vol. 9, pp. 214–223, 2019 (https://doi.org/10.2174/2210327908666181112144939).

[3] S. Buzzi, C. D'Andrea, A. Zappone, and C. D'Elia, "User-centric 5G Cellular Networks: Resource Allocation and Comparison with the Cell-free Massive MIMO Approach", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 1250–1264, 2020 (https://doi.org/10.1109/TWC.2019.2952117).

[4] M. Costa, "Writing on Dirty Paper", *IEEE Transactions on Information Theory*, vol. 29, pp. 439–441, 1983 (https://doi.org/10.1109/TIT.1983.1056659).

[5] M. Mazrouei-Sebdani, W.A. Krzymień, and J. Melzer, "Massive MIMO with Nonlinear Precoding: Large-system Analysis", *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 2815–2820, 2016 (https://doi.org/10.1109/TVT.2015.2425884).

[6] E. Björnson and L. Sanguinetti, "Making Cell-free Massive MIMO Competitive with MMSE Processing and Centralized Implementation", *IEEE Transactions on Wireless Communications*, vol. 19, pp. 77–90, 2020 (https://doi.org/10.1109/TWC.2019.2941478).

[7] A. Björck, *Numeric Methods in Matrix Computations*, Springer, 816 p., 2015 (https://doi.org/10.1007/978-3-319-05089-8).

[8] X. Gao, L. Dai, Y. Ma, and Z. Wang, "Low-complexity Near-optimal Signal Detection for Uplink Large-scale MIMO Systems", *Electronics Letters*, vol. 50, pp. 1326–1328, 2014 (https://doi.org/10.1049/el.2014.0713).

[9] B. Yin, M. Wu, J.R. Cavallaro, and W. Studer, "Conjugate Gradient-based Soft-output Detection and Precoding in Massive MIMO Systems", *IEEE Global Communications Conference (GLOBECOM)*, Austin, USA, 2014 (https://doi.org/10.1109/GLOCOM.2014.7037382).

[10] L. Dai *et al.*, "Low-complexity Soft-output Signal Detection Based on Gauss-Seidel Method for Uplink Multiuser Large-scale MIMO Systems", *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 4839–4845, 2015 (https://doi.org/10.1109/TVT.2014.2370106).

[11] D. Zhu, B. Li, and P. Liang, "On the Matrix Inversion Approximation Based on Neumann Series in Massive MIMO Systems", *IEEE International Conference on Communications (ICC)*, London, UK, 2015 (https://doi.org/10.1109/ICC.2015.7248580).

[12] C. Tang, C. Liu, L. Yuan, and Z. Xing, "High Precision Low Complexity Matrix Inversion Based on Newton Iteration for Data Detection in the Massive MIMO", *IEEE Communications Letters*, vol. 20, pp. 490–493, 2016 (https://doi.org/10.1109/LCOMM.2015.2514281).

[13] C. Zhang *et al.*, "A Low-complexity Massive MIMO Precoding Algorithm based on Chebyshev Iteration", *IEEE Access*, vol. 5, pp. 22545–22551, 2017 (https://doi.org/10.1109/ACCESS.2017.2760881).

[14] M.A. Albreem, M. Juntti, and S. Shahabuddin, "Massive MIMO Detection Techniques: A Survey", *IEEE Communications Surveys & Tutorials*, vol. 21, pp. 3109–3132, 2019 (https://doi.org/10.1109/COMST.2019.2935810).

[15] M.A. Albreem, A.H.A. Habbash, A.M. Abu-Hudrouss, and S.S. Ikki, "Overview of Precoding Techniques for Massive MIMO", *IEEE Access*, vol. 9, pp. 60764–60801, 2021 (https://doi.org/10.1109/ACCESS.2021.3073325).

[16] A. Hadjidimos, "Accelerated Overrelaxation Method", *Mathematics of Computation*, vol. 32, pp. 149–157, 1978.

[17] Y. Hu, J. Wu, and Y. Wang, "SAOR-based Precoding with Enhanced BER Performance for Massive MIMO Systems", *International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, Okinawa, Japan, 2019 (https://doi.org/10.1109/ICAIIC.2019.8668984).

[18] J. Wu, Y. Hu, and Y. Wang, "An Improved AOR-based Precoding for Massive MIMO Systems", *Proc. of 4th International Conference on Communication and Information Processing*, pp. 251–255, 2018 (https://doi.org/10.1145/3290420.3290427).

[19] M.N. Irshad, I.A. Khoso, M.M. Aslam, and M. Silapunt, "Optimized Accelerated Over-relaxation Method for Robust Signal Detection: A Metaheuristic Approach", *Algorithms*, vol. 17, art. no. 463, 2024 (https://doi.org/10.3390/a17100463).

[20] M.N. Irshad, M.M. Aslam, and R. Silapunt, "Low Complexity Optimized AOR Method for Massive MIMO Signal Detection", *IEEE Access*, vol. 13, pp. 51054–51068, 2025 (https://doi.org/10.1109/ACCESS.2025.3550272).

[21] T.L. Marzetta, "Noncooperative Cellular Wireless with Unlimited Numbers of Base Station Antennas", *IEEE Transactions on Wireless Communications*, vol. 9, pp. 3590–3600, 2010 (https://doi.org/10.1109/TWC.2010.092810.091092).

[22] L. Yang *et al.*, "Low-complexity and Fast-convergence Linear Precoding Based on Modified SOR for Massive MIMO Systems", *Digital Signal Processing*, vol. 107, art. no. 102864, 2020 (https://doi.org/10.1016/j.dsp.2020.102864).

[23] N. Aounallah, "Initialization of an Iterative Low-complexity Method for Signal Precoding in MM-Wave Massive MIMO Systems", *Traitement du Signal*, vol. 40, pp. 361–366, 2023 (http://doi.org/10.18280/ts.400136).

[24] A.E. Zorkun, M.A. Salas-Natera, and R.M. Rodriguez-Osorio, "Improved Iterative Inverse Matrix Approximation Algorithm for Zero Forcing Precoding in Large Antenna Arrays", *IEEE Access*, vol. 10, pp. 100964–100975, 2022 (https://doi.org/10.1109/ACCESS.2022.3208155).

[25] E. Bobrov, D. Kropotov, S. Troshin, and D. Zaev, "Study on Precoding Optimization Algorithms in Massive MIMO System with Multi-antenna Users", *Optimization Methods and Software*, vol. 39, pp. 282–297, 2024 (https://doi.org/10.1080/10556788.2022.2091564).

[26] Q. Xue, "The Analysis of the Convergence of the AOR Method and the Comparison with the SOR Method", *Numerical Mathematics A Journal of Chinese Universities*, pp. 39–49, 2006.

[27] C.L. Meng and C.C. Shen, "PSO-based Searching Precoding for MU-MIMO System", *Wireless Personal Communications*, vol. 135, pp. 1845–1860, 2024 (https://doi.org/10.1007/s11277-024-11170-8).

[28] W.B. Abbas, *et al.*, "Heuristic Antenna Selection and Precoding for a Massive MIMO System", *IEEE Open Journal of the Communications Society*, vol. 5, pp. 83–96, 2024 (https://doi:10.1109/OJCOMS.2023.3339402).

[29] A.G. Gad, "Particle Swarm Optimization Algorithm and its Applications: A Systematic Review", *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531–2561, 2022 (https://doi.org/10.1007/s11831-021-09694-4).

Naceur Aounallah, Prof.
Department of Electronic and Telecommunications
https://orcid.org/0000-0001-9137-7900
E-mail: aounallah.naceur@univ-ouargla.dz
Kasdi Merbah Ouargla University, Ouargla, Algeria
https://www.univ-ouargla.dz

Smail Labed, M.Sc.
Electrical Engineering Laboratory (LAGE)
https://orcid.org/0000-0003-1317-8572
E-mail: labed.smail@univ-ouargla.dz
Kasdi Merbah Ouargla University, Ouargla, Algeria
https://www.univ-ouargla.dz

Hybrid Approach for Detection and Mitigation of DDoS Attacks Using Multi-feature Selection, Unsupervised Learning, and Game Theory

Amit Kachavimath and Narayan D.G.

KLE Technological University, Hubballi, Karnataka, India

https://doi.org/10.26636/jtit.2025.4.2261

Abstract — Software-defined networking (SDN) is now widely used in modern network infrastructures, but its centralized control design makes it vulnerable to distributed denial of service (DDoS) attacks targeting the SDN controller. These attacks are capable of disrupting the operation of the network and reducing its availability for genuine users. Existing detection and mitigation methods often suffer from numerous drawbacks, such as high computational costs and frequent false alarms, especially with standard machine learning or basic unsupervised approaches. To address these issues, a new framework is proposed that relies on multistep feature selection methods, including SelectKBest, ANOVA-F, and random forest to select the most important network features, to detect anomalies in an unsupervised manner using agglomerative clustering in order identify suspicious hosts, and to mitigate adverse impacts by relying on posterior probability and game theory. An evaluation conducted using benchmark datasets and validated through Mininet emulation demonstrates that the approach achieves better performance with silhouette scores of 0.86 for InSDN and 0.95 for Mininet. The framework efficiently computes reputation scores to distinguish malicious hosts, thus enabling to rely on adaptive defense against evolving attack patterns while maintaining minimal computational overhead.

Keywords — agglomerative clustering, DDoS attacks, game theory, SDN, unsupervised learning

1. Introduction

The software-defined networking (SDN) concept enhances the process of managing a network by separating the control plane from the data plane and thus enabling centralized data traffic oversight. Such an architecture enables network operators to configure and optimize traffic while using a single control point, simultaneously simplifying policy updates and facilitating the deployment of new services. However, the approach relying on central control comes along with new vulnerabilities. If attackers flood the SDN controller with malicious traffic, with such an attempt known as a distributed denial of service (DDoS) attack, the traffic in the entire network may be slowed down. As more organizations use SDN in cloud and enterprise settings, the detection and mitigation of these attacks is mandatory.

Conventional DDoS detection in SDN often uses fixed rules or simple thresholds to identify malicious traffic. Although these methods are easy to use, they are not flexible enough to handle constantly changing dynamic traffic.

Machine learning and deep learning techniques have shown promise in detecting attacks more accurately, but usually need large amounts of labeled data to train and require significant computing resources [1]. This makes them less practical for real-time implementations in SDNs, especially when attacks take new forms that the model has not experienced before.

Unsupervised learning approaches, such as clustering algorithms, are promising because they do not rely on pre-labeled datasets. These methods are capable of detecting anomalies or suspicious behavior within network traffic by grouping similar patterns, thereby identifying potential attacks. However, the effectiveness of unsupervised detection is critical to the selection of relevant features from network data.

Detection alone is not adequate. Effective mitigation strategies are equally important to ensure the security of SDN environments. Game theory, which analyzes strategic interactions between adversarial entities, offers a valuable framework for improving SDN security. By mathematically modeling the behaviors of both attackers and defenders, game-theoretic methods allow the network controller to dynamically adapt its mitigation policies [2]. This adaptive response improves the controller's ability to manage threats in real time, maintaining an optimal balance between security enforcement and network performance for legitimate users.

The contributions of this research work include the following:

- A combined approach that uses three distinct feature selection methods: SelectKBest, ANOVA F-value and random forest to identify the most relevant network traffic features for effective attack detection.
- Use of agglomerative clustering, i.e. an unsupervised algorithm, to detect anomalous traffic flows that may hint at attacks, even in the absence of labeled attack data.
- Development of a novel game theory-based mitigation process that analyzes traffic flows and applies penalties to suspicious users based on their behavior and puzzle solving speed.

The paper is organized as follows. Section 2 reviews previous work concerning the field in question. Section 3 presents the methodology and modeling details. Section 4 provides an analysis of the experimental findings and Section 5 contains concluding remarks.

2. Related Work

As SDNs become ever more popular, the level of security they offer, especially in defending against DDoS attacks, has become a major area of concern. Legacy techniques, such as statistical analysis and entropy-based models, have been commonly applied for detecting abnormal activities. However, these traditional solutions often face difficulties when it comes to accuracy and do not scale well with large high-speed networks.

In recent years, the use of machine learning and deep learning has become more popular, as these approaches may interpret complex network behaviors and improve detection rates. Research relying on artificial intelligence methods has shown better results in identifying malicious traffic and adapting to changing attack patterns. Despite these advances, some challenges – such as delays in detecting threats, frequent false alarms, and difficulties with managing different types of attacks – still remain. This highlights the need for more adaptable and robust security frameworks for SDN environments.

The authors of [3] propose a clustering framework based on the whale optimization algorithm (WOA-DD) for detecting DDoS attacks in SDNs, using metaheuristic clustering to dynamically group and identify malicious traffic. The methodology separates control and data planes, utilizing programmable SDN switches to analyze network flows and applying WOA-based clustering to detect attack patterns. In [4], the efficiency of an entropy-based technique is identified to detect DDoS attacks on SDN controllers, covering both low- and high-rate attack patterns. The method computes the entropy using the source IP distribution in network traffic, applying a statistical traffic analyzer, and comparing the results with a predefined threshold. The experiments, conducted in a Mininet environment with a POX controller and 64 hosts, evaluate detection rate (DR) and false positive rate (FPR) across eight simulation scenarios, covering both single and multiple attackers targeting one or multiple victims. The results show that the entropy-based approach achieves higher detection rates and lower false positives for high-rate attacks compared to low-rate attacks, with enhancements in DR by 6.25% (to 20.26%) and reductions in FPR by 64.81% (to 77.54%).

The authors of [5] introduced a semi-supervised DDoS detection approach using the K-means clustering algorithm to classify network traffic as normal or malicious. The methodology involves training and testing the model on the CICIDS 2017 dataset, employing hybrid feature selection techniques to optimize input attributes.

Article [6] introduced CAPoW – a context-aware AI-assisted proof-of-work (PoW) framework designed to mitigate DDoS attacks. Using context-aware AI models, such as dynamic attribute-based reputation (DAbR) and temporal activity models (TAM), the framework calculates context scores from deviations in attributes such as IP and temporal data. The CIC-IDS 2017 dataset is used for training and testing, with flow-level features analyzed for adaptive puzzle difficulty. CAPoW achieves a 96% precision level and evaluates performance using such metrics as latency and computational cost, demonstrating its effectiveness in reducing adversarial requests while maintaining network integrity.

In [7], the authors introduce an AI-assisted PoW framework to mitigate DDoS attacks. The methodology uses reputation scores calculated through an AI model trained on the Cisco Talos dataset, which includes malicious IP addresses and attributes. The framework assigns dynamic PoW puzzle difficulties based on reputation scores, integrating Java (Springboot) and Python (Flask) for implementation.

The authors of [8] propose a dynamic game-theoretic framework to mitigate DDoS attacks in SDN-enabled cloud environments. The framework employs the Nash folk theorem to enforce cooperative behavior through reward and punishment mechanisms. Using OpenDaylight controllers, OpenFlow rules, and Snort IDS to detect malicious activity, the solution reduced attack traffic by over 90% during evaluations in Mininet with ICMP, TCP SYN, and UDP flood attacks, while maintaining legitimate traffic throughput.

In [9], a cost-effective shuffling algorithm (CES) is proposed within a moving target defense (MTD) framework to counter DDoS attacks. The methodology utilizes multiobjective Markov decision processes (MOMDP) to model interactions between attackers and defenders, enabling dynamic shuffling decisions. The CES algorithm was implemented using OpenStack and Open vSwitch in an SDN testbed with 50 virtual machines.

The authors of [10] propose an autonomous DoS/DDoS defense system for SDN networks, called GT-HWDS. It combines the Holt-Winters for digital signature (HWDS) and the game-theoretical (GT) approaches. The methodology includes seven-dimensional IP flow analysis for anomaly detection and a game theory-based mitigation module for optimal defense strategies. Using real IP flow data from a large-scale network and the Scorpius simulator for attack scenarios, the system achieves an anomaly detection accuracy of over 98%.

In [11], the authors propose a game-theoretical approach to mitigate DDoS attacks in edge computing environments, focusing on edge DDoS mitigation (EDM). The methodology introduces two approaches: EDMOpti for small-scale optimization using integer programming and EDMGame for large-scale suboptimal solutions employing Nash equilibrium. Evaluations conducted using the EUA data set and simulated edge server scenarios demonstrate the ability to reduce service latency and effectively mitigate over 90% of malicious traffic.

A security enforcement framework aimed at strengthening SDN controllers through game-theoretic defense models against various attack vectors is proposed in [12]. The ap-

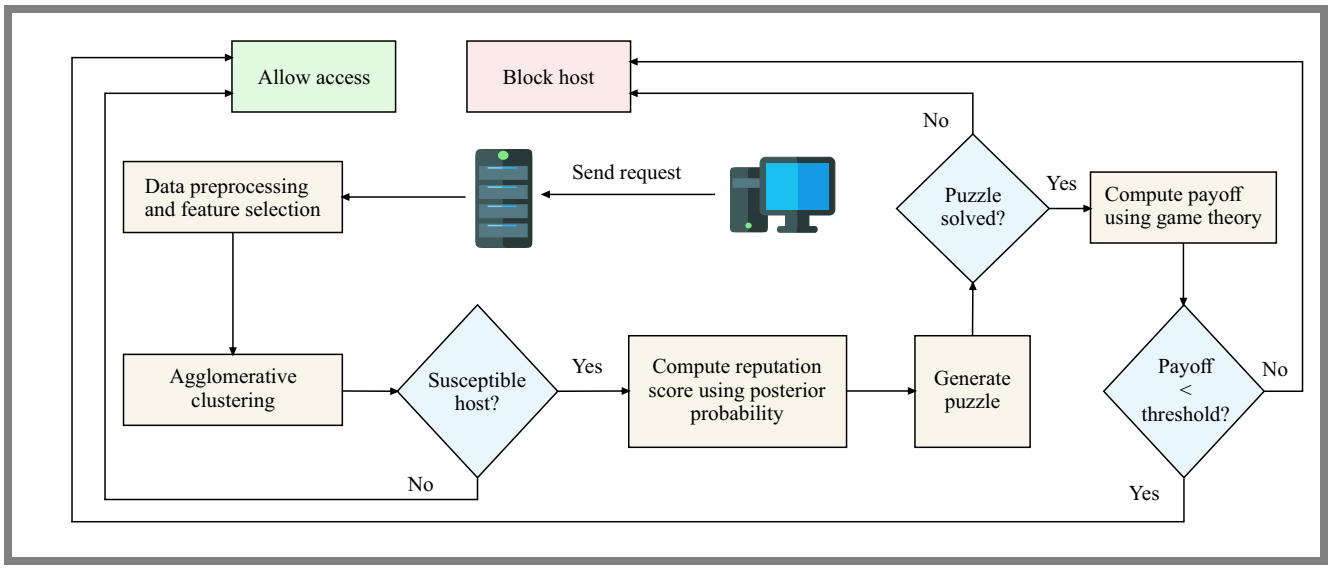

Fig. 1. Architecture of the proposed system.

proach incorporates a trust-based detection scheme that uses signaling game principles for early identification of threats, along with a risk-oriented prevention mechanism to assess and control threat levels in packet flows. To evaluate its effectiveness, the framework models potential threats using the STRIDE methodology and performs experiments in Mininet, achieving a detection accuracy of 98%.

In [13], the authors present a dynamic defense strategy to counter volumetric DDoS attacks by employing client puzzles as a PoW mechanism. The proposed framework adopts a multi-tier approach, incorporating a traffic analysis unit, a puzzle generation component, and a dynamic provisioning module to regulate and suppress attack traffic, with the evaluation conducted using the NS2 simulator.

A zero-sum game-based anti-DDoS firewall framework is proposed to mitigate DDoS attacks in [14]. To optimize mitigation strategies, the framework models interactions between attackers and defenders as a linear programming problem. Two mitigation algorithms are introduced, employing connection limits and sending rates to manage traffic on 80/443 web ports. It is validated using a pay-off matrix, demonstrating probabilistic decision making for effective defense. The study described in [15] proposes a hybrid defense mechanism designed to mitigate distributed reflection denial-of-service attacks within 5G network environments. The framework utilizes a Stackelberg game model to optimize packet sampling rates, combining SDN detection with target-side defenses.

The authors of [16] propose a PoW mechanism integrated with the game theory to mitigate DDoS attacks using computational puzzles with dynamic difficulty levels based on the request frequency of clients. Simulations using 2 000 clients, including 1 000 legitimate users and 1 000 scalpers, demonstrate the system's ability to limit malicious traffic while ensuring that legitimate users are capable of accessing more resources.

In [17], the authors present a game-theoretic framework for identifying and countering low-rate denial-of-service (LR-DoS) attacks. Their approach employs a sigmoid-based filter to distinguish between legitimate and malicious traffic using bandwidth thresholds, while directing potentially harmful flows towards honeypot systems for further analysis. The game theory model evaluates strategies of both attackers and defenders, identifying optimal actions, and achieving Nash equilibrium to enhance security. It uses the NS-3 simulator for traffic generation and Matlab for analysis.

A dynamic Bayesian game-based approach to safeguard software-defined space-air-ground integrated networks (SD-SAGIN) from DDoS attacks is introduced in [18]. This method utilizes support vector machines (SVM) for detecting attacks and leverages Nash equilibrium to dynamically optimize strategies for both attackers and defenders and is validated through simulations conducted in Mininet using a Ryu controller.

All the related works highlight significant gaps in DDoS mitigation strategies for SDN. Traditional methods, such as entropy-based and statistical detection approaches lack the accuracy and scalability needed for modern high-traffic networks. Machine learning and deep learning techniques have shown promise but face challenges, including high false positives, detection latency, and handling heterogeneous attack types. Existing game theory and PoW-based methods improve mitigation, but often involve excessive computational overhead or fail to adapt dynamically to evolving attack scenarios. These limitations emphasize the need for innovative, scalable, and adaptive solutions that balance detection efficiency with resource optimization.

3. Proposed Methodology

The proposed system model, illustrated in Fig. 1, intended for mitigating DDoS attacks in SDN environments, is initiated by

continuously monitoring real-time network traffic and capturing each incoming flow for analysis. During preprocessing, relevant traffic attributes, such as source IP and protocol information, are extracted and normalized. A comprehensive feature selection process is then performed using multiple statistical techniques. SelectKBest, ANOVA-F, and random forest are relied upon to identify the most significant features for downstream analysis. These selected attributes are input to an unsupervised anomaly detection module, where agglomerative clustering is applied to distinguish between normal and suspicious network activities.

If abnormal traffic is detected, the system proceeds to assess the reputation score and dynamically assigns a puzzle difficulty threshold based on posterior probabilities and statistical characteristics of the traffic. A game-theoretic mitigation strategy is used as the model constructs a pay-off matrix and calculates expected utilities to guide response decisions. The final decision module implements mitigation measures, such as blocking or isolating the malicious source, while allowing legitimate traffic to proceed. This adaptive and multilayered approach enhances detection accuracy and ensures efficient mitigation in real-time programmable network environments.

The proposed framework operates in a periodic monitoring cycle to enable continuous detection and mitigation of malicious activities. A one-time feature selection stage is executed at initialization to identify the most relevant flow attributes which are then reused across subsequent intervals. During each monitoring window, the system executes three tasks: unsupervised clustering of captured flows, reputation scoring of active sources, and assignment of proof-of-work puzzle difficulties. The mitigation rules derived from these steps are enforced by the controller in the subsequent cycle. This design avoids redundant recomputation, ensuring that only clustering, scoring, and mitigation are repeated. In our experiments, each cycle required approximately 3 to 5 s to complete, confirming the feasibility of real-time deployment in SDN environments.

3.1. Traffic Capture Using Wireshark

In the proposed framework, real-time network traffic is continuously monitored at the SDN controller using Wireshark. Each monitoring session is configured to capture packets at fixed intervals of 120 s, with the resulting traffic data stored in `pcap` format to maintain detailed records for analysis. After data collection, categorical attributes such as protocol types are converted into unique numerical representations, and all features are standardized to integer or floating-point types for consistency in subsequent analysis.

The set of packets captured during each interval is denoted as $P = \{p_1, p_2, \ldots, p_m\}$, where m represents the total number of packets observed in a single 120-second window. Feature extraction is applied to this set, and the ten most informative attributes are selected, resulting in a feature subset $F_{10} = \{f_1, f_2, \ldots, f_{10}\}$ for downstream anomaly detection. To improve computational efficiency and focus the analysis on potentially abnormal events, the system introduces a packet rate filter for all observed source IP addresses. For each capture interval, the mean packet rate is computed across all unique sources. This is mathematically defined as:

$$\frac{1}{N}\sum_{i=1}^{N} R_i < 100\ , \tag{1}$$

where N denotes the number of distinct source IP addresses present in the interval, and R_i is the packet rate corresponding to the i-th source IP.

If the average packet rate across all sources is less than 100 packets per second (pps), the interval is considered to represent normal traffic and is excluded from clustering and further analysis. This approach ensures that the anomaly detection module processes only intervals with significant activity, which may indicate the onset of a DDoS attack.

The specific selection of 100 pps as a threshold is based on empirical studies and baseline measurements of typical network usage patterns in SDN environments. Under normal conditions, genuine sources usually remain within this rate, whereas DDoS attacks often show sudden increases in the number of packets sent. By setting the threshold at 100 pps, the model effectively filters out benign fluctuations and minimizes false positives, thereby enhancing the detection accuracy and focusing computational resources on suspicious events.

Algorithm 1 Preprocessing network traffic

```
Input: List of pcap files
Output: Numerical feature matrix X
for all file p in pcap files do
    Extract flows: F ← CICFlowMeter(p)
    for all flow f in F do
        Extract features: v ← Features81(f)
        Add v to feature list V
    end for
end for
for all feature c in V do
    if c is categorical then
        c ← LabelEncode(c)
    end if
end for
X ← Convert V to numerical matrix
return X
```

3.2. Data Pre-processing

The data preprocessing phase begins after network traffic is captured using Wireshark in pcap format, initiating a structured workflow to prepare data for unsupervised machine learning detection – see Algorithm 1. The raw pcap files are first processed by CICFlowMeter, which extracts 81 comprehensive flow-based features from each network session, encompassing statistical, temporal, and protocol-specific characteristics of the observed traffic. These features provide a detailed view of typical and potentially abnormal behaviors in the network.

To ensure compatibility with machine learning algorithms that require numerical input, all categorical variables within the feature set are systematically transformed into unique numeric

values using label encoding. This standardization process produces a uniform numerical feature matrix, facilitating effective anomaly detection by unsupervised models.

3.3. Feature Selection

The feature selection process shown as Algorithm 2 is a combination of three well-known methods deloyed to find the most relevant features from data set D, consisting of 81 variables and a target variable y. The aim is to identify ten most important features of F_{top} that contribute the most to distinguishing different types of network traffic.

Algorithm 2 Multi-method feature selection

Input: Dataset D with 81 features, target variable y
Output: Top 10 selected features F_{top}
Initialize $k \leftarrow 20$, $m \leftarrow 10$
$S \leftarrow$ empty list ▷ Store selected features
for all method ∈ {SelectKBest, ANOVA F-value, random forest} **do**
  if method = SelectKBest **then**
    Compute statistical scores for each feature
    Select top k features; add to S
  else if method = ANOVA F-value **then**
    Compute F-value for each feature
    Select top k features; add to S
  else if method = random forest **then**
    Train random forest classifier
    Rank features by importance
    Select top k features; add to S
  end if
end for
Initialize $freq$ as an empty map ▷ Count feature ▷ occurrences
for all feature f in S **do**
  if f in $freq$ **then**
    $freq[f] \leftarrow freq[f] + 1$
  else
    $freq[f] \leftarrow 1$
  end if
end for
Sort features in $freq$ by frequency (descending)
$F_{top} \leftarrow$ Top m features with highest frequency
return F_{top}

First, the algorithm sets $k = 20$, meaning that each method will select its top 20 features, and $m = 10$, i.e. the number of final features to be chosen. An empty list S is used to collect all the features selected by each method. The algorithm evaluates three feature selection techniques in sequence: SelectKBest, ANOVA F-value, and random forest. For SelectKBest, the method performs a statistical test for each feature in D and selects 20 features with the highest test scores, adding them to S. The ANOVA F-value method calculates how well each feature separates the classes in y and also picks the top 20 features.

The random forest method builds a classifier using all features and calculates an importance score for each feature.

Tab. 1. Top 10 selected features from the InSDN dataset.

No.	Name	Description of feature
1	Src IP	Network address from which the data originates
2	Dst IP	Destination address to which the data is directed
3	Protocol	Specifies the type of network protocol, e.g. TCP or UDP
4	Pkt Len Std	Standard deviation of all packet sizes within a flow
5	Flow ID	Unique identifier allocated to each network flow
6	Bwd Pkt Len mean	Average size of packets sent in the reverse direction
7	Pkt Len Var	Measure of how packet sizes vary within a single flow
8	Src port	Port number used by the sender of the packet
9	Bwd Seg Avg size	Average segment size calculated for the backward path
10	Bwd Pkt Std Len	Standard deviation of packet lengths in the backward direction

Again, the 20 highest-ranked features are added to S. Once all three methods have selected their features, the algorithm counts how often each feature appears in the combined list S using a frequency map. Features that appear more frequently across the various methods are considered more consistently important.

The algorithm then sorts all features by their frequency, in descending order, and selects the top m features as F_{top}. This approach ensures that the final selection is not biased toward any single method and that only features considered important by multiple techniques are retained. Then, ten features in F_{top} are used in the subsequent steps of anomaly detection and model development process, helping to improve both detection accuracy and computational efficiency (Tab. 1).

3.4. Unsupervised Learning

The proposed Algorithm 3 describes an unsupervised anomaly detection procedure relying on hierarchical agglomerative clustering and subsequent evaluation of cluster quality using the silhouette score.

The input to the algorithm is a feature set $X = x_1, x_2, \ldots, x_m$, where x_i denotes a data point in the selected feature space. Before clustering, the algorithm enforces two prerequisites: network traffic must be aggregated over a 120-second interval, and the average packet rate per flow must exceed 100 pps. This ensures that only sufficiently active and potentially anomalous traffic segments are subjected to further analysis.

If these conditions are satisfied, agglomerative clustering is applied to X to partition the data into two clusters, corresponding to normal and suspicious traffic. The result of

the clustering assigns label $c_i \in 0, 1$ to each data point x_i, resulting in the cluster assignment set $C = c_1, c_2, \ldots, c_m$.

Equation (2) represents the mean intracluster distance a_i for a data point x_i. It is obtained by computing the average Euclidean distance between x_i and every other member of its own cluster C_{xi}, excluding the point itself:

$$a_i = \frac{1}{|C_{x_i}| - 1} \sum_{x_j \in C_{x_i},\, x_j \neq x_i} d(x_i, x_j) \,, \tag{2}$$

where $|C_{x_i}|$ denotes the total number of points in the cluster C_{x_i} and $d(x_i, x_j)$ represents the Euclidean distance between the points x_i and x_j.

Equation (3) defines the mean distance b_i for x_i. This is determined by locating the cluster C_k such that $C_k \neq C_{x_i}$ and is closest to x_i, then computing the average distances from x_i to every point within C_k:

$$b_i = \min_{C_k \neq C_{x_i}} \frac{1}{|C_k|} \sum_{x_j \in C_k} d(x_i, x_j) \,. \tag{3}$$

The silhouette score s_i for a data point x_i is computed as the difference between the mean distance from x_i to the closest neighboring cluster and the mean distance to points within its own cluster, scaled by the larger of these two quantities:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} \,. \tag{4}$$

This score indicates how similar x_i is to its own cluster relative to the nearest neighboring cluster. The overall silhouette score S for the entire data set as the mean of all individual silhouette scores is:

$$S = \frac{1}{m} \sum_{i=1}^{m} s_i \,. \tag{5}$$

A higher value of S (close to 1) suggests well-defined, separate clusters. Values near zero indicate overlapping or ambiguous clusters, and negative values may suggest misclassified points. The algorithm ultimately outputs both cluster assignments C and the computed silhouette score S, which form the basis for further traffic classification and model validation.

3.5. Computation of Reputation Score

The prior probability that the traffic from source IP IP_i is:

$$P(\text{Normal} \mid IP_i) = \frac{n_0(IP_i)}{n_{IP_i}} \,, \tag{6}$$

where $n_0(IP_i)$ is the count of normal traffic samples and n_{IP_i} is the total number of samples from IP_i within a monitoring window. This value serves as the initial baseline for further traffic classification.

Equation (7) defines the prior probability that the traffic from IP_i is malicious:

$$P(\text{Malicious} \mid IP_i) = \frac{n_1(IP_i)}{n_{IP_i}} \,, \tag{7}$$

where $n_1(IP_i)$ is the number of malicious samples from IP_i. This value helps to estimate the likelihood of attack activity from each source.

Algorithm 3 Agglomerative clustering and silhouette score evaluation

Input: Feature set $X = \{x_1, x_2, \ldots, x_m\}$
Output: Cluster assignments C, silhouette score S
Check clustering conditions:
if traffic interval = 120 s **and** avg. flow packets/s > 100 **then**
  Proceed to clustering
else
  Exit (do not cluster)
end if
Apply agglomerative clustering on X to form two clusters $(n = 2)$
Obtain cluster labels $C = \{c_1, c_2, \ldots, c_m\}$, $c_i \in \{0, 1\}$
for each data point x_i in X **do**
  Calculate intra-cluster distance:

$$a_i = \frac{1}{|C_{x_i}| - 1} \sum_{x_j \in C_{x_i},\, x_j \neq x_i} d(x_i, x_j)$$

  Calculate nearest-cluster distance:

$$b_i = \min_{C_k \neq C_{x_i}} \frac{1}{|C_k|} \sum_{x_j \in C_k} d(x_i, x_j)$$

  Compute silhouette score for x_i:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

end for
Compute overall silhouette score:

$$S = \frac{1}{m} \sum_{i=1}^{m} s_i$$

return cluster labels C, silhouette score S

The mean μ_{Normal} and standard deviation σ_{Normal} for a feature x among all samples labeled normal ($c_i = 0$) for a specific IP are defined as follows. These statistics summarize the typical behavior of normal flows.

$$\mu_{\text{Normal}} = \frac{1}{n_0^{(j)}} \sum_{i:c_i=0} x_i \,, \tag{8}$$

$$\sigma_{\text{Normal}} = \sqrt{\frac{1}{n_0^{(j)}} \sum_{i:c_i=0} (x_i - \mu_{\text{Normal}})^2} \,. \tag{9}$$

The mean value $\mu_{\text{Malicious}}$ for a particular feature x is calculated on all data samples labeled as malicious for a specific source IP.

$$\mu_{\text{Malicious}} = \frac{1}{n_1^{(j)}} \sum i : c_i = 1 x_i \,, \tag{10}$$

where $n_1^{(j)}$ is the total count of malicious samples for IP j.

The standard deviation $\sigma_{\text{Malicious}}$ for the same feature x provides the spread of feature values around the malicious mean $\mu_{\text{Malicious}}$ for samples with $c_i = 1$. This statistic helps capture the variability of the selected feature among flows classified

as attacks for the particular IP.

$$\sigma_{\text{Malicious}} = \sqrt{\frac{1}{n_1^{(j)}} \sum i : c_i = 1 (x_i - \mu_{\text{Malicious}})^2} \,. \tag{11}$$

The probability of observing feature value x given class-specific mean μ and standard deviation σ under a Gaussian distribution is modeled in following manner. This likelihood supports probabilistic classification for each flow.

$$P(x \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(- \frac{(x-\mu)^2}{2\sigma^2} \right) . \tag{12}$$

The posterior probability that a flow with features x belongs to a specific class, integrating both likelihood and prior probability using Bayes' theorem, is defined as:

$$P(\text{Class} \mid x) = \frac{P(x \mid \mu_{\text{Class}}, \sigma_{\text{Class}}) \cdot P(\text{Class})}{P(x)} \,. \tag{13}$$

The normalization constant $P(x)$ ensuring that the posterior probabilities across all classes sum up to one is:

$$\begin{aligned} P(x) = {} & P(x \mid \mu_{\text{Normal}}, \sigma_{\text{Normal}}) \cdot P(\text{Normal}) \\ & + P(x \mid \mu_{\text{Malicious}}, \sigma_{\text{Malicious}}) \cdot P(\text{Malicious}) \,. \end{aligned} \tag{14}$$

The reputation score R for a flow, scaling the posterior malicious probability to a score between 0 and 10, which influences subsequent mitigation actions, is defined by the following:

$$R = 10 \cdot P(\text{Malicious} \mid x) \,. \tag{15}$$

Equation (16) assigns the puzzle difficulty D according to the calculated malicious probability. Higher risk results in a more challenging proof-of-work puzzle for the source.

$$D = \begin{cases} 5 \text{ (Very high)}, & P(\text{Malicious} \mid x) > 0.8 \\ 4 \text{ (High)}, & 0.6 < P(\text{Malicious} \mid x) \leqslant 0.8 \\ 3 \text{ (Medium)}, & 0.4 < P(\text{Malicious} \mid x) \leqslant 0.6 \\ 2 \text{ (Low)}, & 0.2 < P(\text{Malicious} \mid x) \leqslant 0.4 \\ 1 \text{ (Very low)}, & P(\text{Malicious} \mid x) \leqslant 0.2 \end{cases} \tag{16}$$

3.6. Mitigation using PoW and Game Theory

The construction of the hash input for puzzle generation, concatenating flow attributes, and a nonce value before SHA-256 computation, is formulated as:

$$\begin{aligned} H = {} & \text{SHA-256 (Src IP} \mid \text{Protocol} \mid \text{Flow duration} \\ & \mid R \mid \text{Nonce)} \,. \end{aligned} \tag{17}$$

The condition for puzzle completion is stated in this way. The hash output H must begin with D consecutive zeros, signifying successful proof-of-work.

$$H[1:D] = \text{"0} \ldots \text{0"} \ (D \text{ leading zeros}) \,. \tag{18}$$

The adaptive time threshold for solving the puzzle is modeled by the following formula which integrates puzzle difficulty D, current network load N_{load}, average solving time $T_{\text{normal avg}}$, and reputation score R weighted by coefficients $\alpha, \beta, \gamma, \delta$.

$$T_{\text{threshold}} = \alpha D + \beta N_{\text{load}} + \gamma T_{\text{normal avg}} - \delta R \,. \tag{19}$$

Tab. 2. Description of the notation used.

Symbol	Description
p_i	Individual packet in the captured traffic
F	Set of all flows extracted from traffic
f_j	A single flow in the dataset
$X = \{x_1, x_2, \ldots, x_m\}$	Feature set of m data points (flows) used for clustering
m	Total number of packets observed in the interval
k	Number of features selected per method during feature selection
$F_{10} = \{f_1, f_2, \ldots, f_{10}\}$	Reduced set of ten most informative features selected from the packet stream
a_i	Mean intra-cluster distance for data point x_i
b_i	Mean nearest-cluster distance for data point x_i
s_i	Silhouette score for data point x_i
S	Overall silhouette score for clustering
$P(\text{Normal} \mid IP_i)$	Probability that traffic from source IP_i is normal
$P(\text{Malicious} \mid IP_i)$	Probability that traffic from source IP_i is malicious
R	Reputation score of a source (range 0 – 10)
D	Puzzle difficulty level assigned to a source
H	Hash value used in proof-of-work puzzle generation
N	Total number of unique source IP addresses identified in the interval
R_i	Packet rate for the i-th source IP address, measured in pps
N_{load}	Current network load during puzzle assignment
T_{solve}	Actual puzzle solving time for a source host
$T_{\text{threshold}}$	Maximum allowed puzzle solving time based on system parameters
U_{A1}, U_{A2}	Expected utility values for attacker strategies in the game-theoretic model

The host status is determined in this way. If the puzzle is solved within the threshold time, the host is normal. Otherwise, it is

flagged as malicious.

$$T_{\text{solve}} \leqslant T_{\text{threshold}} \implies \text{Normal Host}\,, \tag{20}$$

$$T_{\text{solve}} > T_{\text{threshold}} \implies \text{Malicious Host}\,. \tag{21}$$

The expected utility for the host, if it attempts to solve the puzzle, considering both correct S_1 and incorrect S_2 system classifications, is computed as:

$$U_{A_1} = P(S_1) \cdot 10 + P(S_2) \cdot (-5)\,. \tag{22}$$

The expected utility for the host, if it chooses not to solve the puzzle, is once again based on the system classification outcomes. These utility functions inform the optimal strategy and support the mitigation decision.

$$U_{A_2} = P(S_1) \cdot 0 + P(S_2) \cdot (-20)\,. \tag{23}$$

Table 2 provides a consolidated overview of the mathematical notation used throughout the study. It describes the symbols used to represent packets, flows, feature sets, clustering measures, and reputation scores, thus ensuring consistency in the formulation of equations.

4. Results and Discussion

Mininet is used to emulate the SDN environment, allowing for the creation of a programmable network topology for realistic testing. To simulate DDoS behavior, Hping3 is deployed to generate legitimate and attack traffic.

The POX controller, an open source SDN controller, manages network flows, applies predefined rules, and processes packets adaptively. Wireshark captures network traffic, offering a detailed view of packet behavior and transmission patterns. CICFlowMeter is then applied to extract flow-level statistics which are later used to select key features for the detection algorithm. The integration of these tools facilitates continuous monitoring, traffic classification, and dynamic response to network anomalies.

4.1. Analysis of Benchmark Datasets

The CICDDoS 2019 dataset, created by the Canadian Institute of Cybersecurity, provides a collection of real and attack traffic across various DDoS types such as TCP, UDP, SYN flood, and HTTP flood, as represented in Tab. 3. Captured over several days in a controlled environment, it includes labeled flow data generated using CICFlowMeter, with more than 80 features extracted per flow. This structure supports the development and evaluation of both supervised and unsupervised detection models. Due to its diversity and scale, the data set is widely used in SDN security research to benchmark anomaly detection techniques [19].

The InSDN dataset was generated within a realistic SDN architecture, featuring a layered topology that separates data, control, and application planes, as shown in Tab. 4. The testbed includes OpenFlow enabled switches, a centralized POX controller, and a range of hosts that act as both normal users and attackers. Traffic traces were captured from multiple points in the network to reflect operational conditions. The data set comprises labeled flow records that distinguish between normal, DDoS, and other intrusion events, offering more than 80 statistical features per flow using CICFlowMeter. Attack scenarios include volumetric floods and controller-targeted exploits, all systematically varied in terms of their intensity and duration [20].

Tab. 3. CICDDoS 2019 dataset summary.

Parameter	Value
Total duration	5 days
Total packets	50 million+
Attack types	12
Features extracted	84 flow-based attributes
Flow extraction tool	CICFlowMeter
Data format	PCAP and CSV
Label availability	Benign / attack
Traffic type	Normal and malicious

Tab. 4. InSDN dataset summary.

Parameter	Value
SDN topology	Layered (3 planes)
Controller	POX (centralized)
Switch type	OpenFlow-enabled
Attack types	DDoS (SYN, UDP, ICMP), others
Total DDoS flows	291 330
Total DDoS packets	27 million
Feature extraction	CICFlowMeter (80+ features)
Label classes	Normal, DDoS, others
Traffic capture	Multi-point, real/attack flows

The CICIoT 2023 data set was collected in a realistic smart IoT lab environment using 105 diverse IoT devices, including cameras, smart plugs, lights, and home automation controllers, as illustrated in Tab. 5.

Data gathering focused on both benign traffic and malicious activity, especially on DDoS attacks which were launched

Tab. 5. CICIoT 2023 dataset summary.

Parameter	Value
Environment	Smart IoT Lab (real devices)
Number of devices	105 IoT devices
Attack types	DDoS attacks
Total DDoS rows	33 million
Traffic format	PCAP (Wireshark), CSV
Feature extraction	Flow-based, 50+ features
Label classes	Normal, DDoS, other attacks
Attack tools	Raspberry Pi Bots, scripts
Traffic capture	Network tap

by grouping multiple Raspberry Pi devices used as attackers. Various types of DDoS attacks, such as UDP flood, TCP flood, SYN flood, ICMP flood, and more advanced threats such as ACK fragmentation and PSH-ACK flood, were executed to target IoT devices. Network traffic was captured using Wireshark in the pcap format via network taps that mirrored all communication flows without impacting normal operations. After collection, the data were processed into CSV files with their features extracted, resulting in a data set that covers 33 different attacks in seven categories and totaling over 33 million DDoS attack rows [21].

4.2. Mininet Setup Environment

The network is emulated using Mininet (as show in Fig. 2) – a platform that is widely adopted for creating and testing programmable network topologies in a controlled environment. The POX controller is used to manage flow rules and dynamically control network behavior throughout the experiment. The attached network represents a tree topology with the POX controller at the root, connected to core switches S1 and S2, which branch out to the edge switches S3, S4, S5 and S6, and finally connect to end hosts, with their designations ranging from H1 to H100.

This hierarchical structure is commonly used in SDN experiments to model scalable, realistic aggregation, and distribution of network traffic. The raw network traffic was initially captured in pcap format using Wireshark to ensure comprehensive recording of all packets.

For the experiments, the range of IP addresses assigned to the end hosts ranged from 10.0.0.1 to 10.0.0.100. Each host within the topology was uniquely identified by its respective IP address, ensuring clear mapping and traceability of network flows during traffic generation and analysis. This configuration enabled a precise evaluation of both normal and attack scenarios across the network. The HPing3 tool was used to generate both normal and attack traffic in the experimental setup.

Normal flows simulated routine host communications, while malicious traffic included various DDoS patterns such as TCP SYN, UDP, and ICMP floods. The intensity and timing were varied to test the detection system under various network conditions. The resulting traffic was used to rigorously evaluate the performance of the proposed detection framework.

Wireshark was used to capture the pcap files. These were then processed by CICFlowMeter to extract detailed flow-level features, resulting in CSV files suitable for machine learning analysis. During the data preprocessing stage, label encoding was applied to convert categorical variables to numerical values, and any missing or duplicate entries were systematically removed to maintain data quality.

To enhance the effectiveness of subsequent anomaly detection, a feature selection process was conducted, narrowing the data set to the ten most relevant attributes. The refined dataset was then used for unsupervised clustering to identify anomalous traffic patterns. The silhouette score is a metric that is widely used for evaluating the quality of clustering in unsupervised

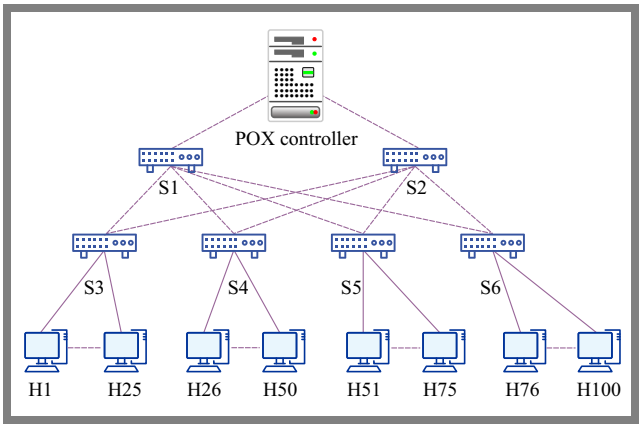

Fig. 2. Mininet topology with 100 hosts.

Tab. 6. Performance evaluation of the silhouette score.

Dataset	Agglomerative	K-means	IF
CICDDoS 2019	0.7084	0.6016	0.5393
InSDN	0.8658	0.3579	0.4203
CICIoT 2023	0.8253	0.2669	0.4114
Mininet	0.9558	0.8023	0.2473

learning. It is computed for each data point as the difference between the mean nearest-cluster distance and the mean intra-cluster distance, divided by the maximum of these two values:

$$s_i = \frac{b_i - a_i}{\max(a_i, b_i)} , \tag{24}$$

where a_i represents the average distance between a point and all other points in its assigned cluster and b_i denotes the lowest average distance to points in any other cluster.

The overall score is the average of s_i across all data points. A silhouette score close to 1 indicates that clusters are well separated and compact, while values near zero suggest overlapping or ambiguous clusters. Negative values indicate potential misclassification. Therefore, higher silhouette scores signify more effective clustering.

Table 6 and Fig. 3 present the silhouette scores obtained for different algorithms, including agglomerative clustering, K-means, and isolation forest with three benchmark datasets and real-time emulation using Mininet. Agglomerative clustering consistently achieves the highest silhouette scores in all datasets, with values of 0.7084 for CICDDoS 2019, 0.8658 for InSDN, 0.8253 for CICIoT 2023, and 0.9558 for the Mininet emulation.

In contrast, K-means and isolation forest yield significantly lower scores, particularly in the case of InSDN and CICIoT 2023 datasets. Consistently high silhouette scores achieved by the proposed approach further validate its effectiveness for real-time DDoS attack detection and traffic analysis.

The Davies-Bouldin (DB) index is a metric that is commonly used to assess clustering performance, as it measures the average similarity between each cluster and its most similar cluster. It is defined as the ratio of scatter to between-cluster separation between clusters for all clusters. Lower DB in-

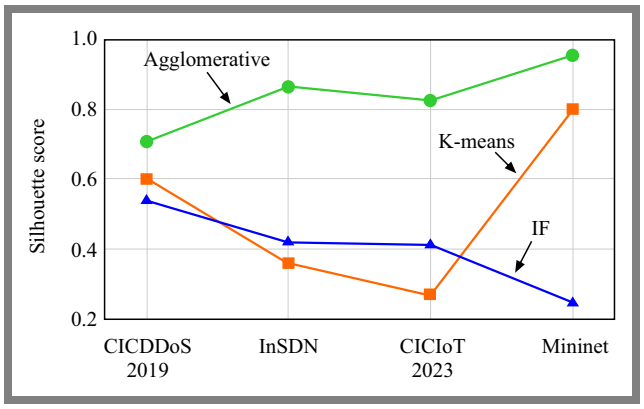

Fig. 3. Performance evaluation using silhouette score.

Tab. 7. Performance evaluation of Davies-Bouldin index scores.

Dataset	Agglomerative	K-means	IF
CICDDoS 2019	0.5139	0.7809	2.9820
InSDN	0.0951	1.4336	2.3158
CICIoT 2023	0.2932	1.8647	3.0906
Mininet	0.0392	0.1766	3.0786

dex values indicate better clustering as they reflect compact clusters with greater separation from each other.

The DB index for a clustering solution with k clusters is calculated as follows:

$$DB = \frac{1}{k}\sum_{i=1}^{k}\max_{j\neq i}\left(\frac{S_i + S_j}{M_{ij}}\right), \tag{25}$$

where S_i and S_j are the average distances between all points in i and j and their respective centroids, and M_{ij} is the distance between the centroids of clusters i and j.

Table 7 and Fig. 4 present the Davies-Bouldin index scores for agglomerative clustering, K-means, and isolation forest approaches. Agglomerative clustering consistently achieves the lowest values of the DB index, with scores of 0.5139 for CICDDoS 2019, 0.0951 for InSDN, 0.2932 for CICIoT 2023, and 0.0392 for Mininet. In comparison, K-means and isolation forest yield higher DB index values, reflecting less compact and poorly separated clusters, particularly for the IoT and InSDN datasets. The consistently low DB index achieved by agglomerative clustering highlights its effectiveness for DDoS attack detection in SDN and IoT environments.

The Calinski-Harabasz (CH) index, also known as the variance ratio criterion, is a standard metric relied upon for evaluating clustering performance. It is defined as the ratio of between-cluster dispersion to within-cluster dispersion:

$$CH = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \cdot \frac{n-k}{k-1}, \tag{26}$$

where $\mathrm{Tr}(B_k)$ is the trace of the dispersion matrix, $\mathrm{Tr}(W_k)$ is the trace of the dispersion matrix, n is the number of samples and k is the number of clusters.

Higher CH index values indicate better clustering with well-separated and compact groups.

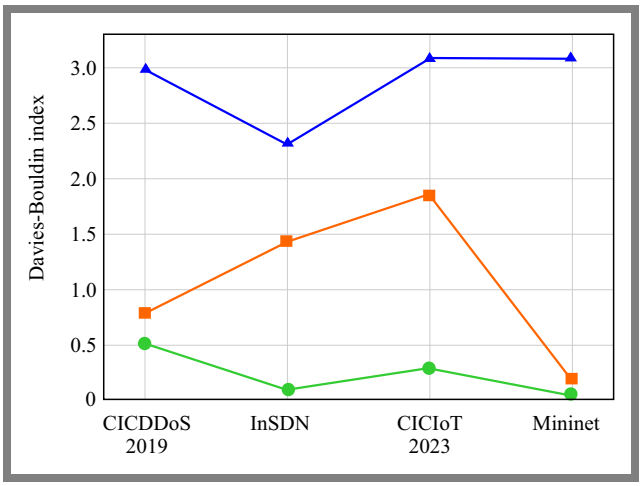

Fig. 4. Performance evaluation using Davies-Bouldin index.

Tab. 8. Performance evaluation of Calinski-Harabasz index scores.

Dataset	Agglomerative	K-means	IF
CICDDoS 2019	9832.68	7402.58	718.94
InSDN	56297.22	3443.00	1074.22
CICIoT 2023	264.94	1816.02	528.59
Mininet	13902.40	84479.42	129.07

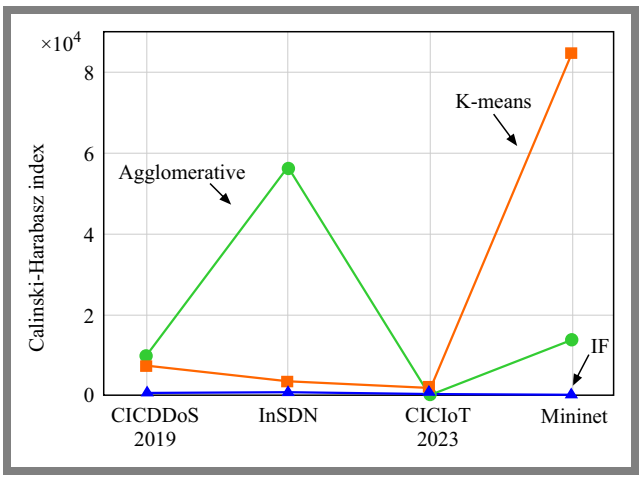

Fig. 5. Performance evaluation using the Calinski-Harabasz index.

Table 8 and Fig. 5 show that agglomerative clustering achieves high CH index scores across most datasets, especially in InSDN and Mininet, suggesting strong cluster separation and compactness.

Figure 6 presents the distribution of the reputation scores assigned during the detection process. The histogram shows that the majority of traffic flows are concentrated around reputation scores between 9 and 10, with the highest frequencies observed being close to 9.5. This pattern indicates that most observed flows represent malicious traffic, reflecting the prevalence of benign or ambiguous traffic in the dataset.

Only a small number of flows receive reputation scores below 9. The spread and shape of the distribution support the effectiveness of the reputation scoring mechanism, proving its

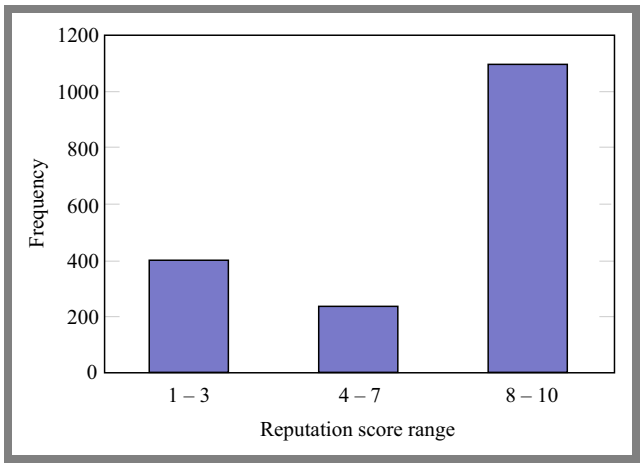

Fig. 6. Distribution of reputation scores (grouped ranges).

Tab. 9. Simulation parameters used to analyze the proposed system.

Parameter	Description	Value
Number of clusters k	Partitioning of dataset into groups	2
Linkage criterion	Rule for merging clusters	Ward
Affinity / Distance metric	Distance computation method	Euclidean
Compute full tree	Construction of complete hierarchy	True
Compute distances	Store distances between merged clusters	True
Silhouette score	Measure of clustering quality	0.95

effectiveness in distinguishing between typical background traffic and potential threats in the SDN environment.

Table 9 summarizes the hyperparameter settings for agglomerative clustering. The configuration with $k = 2$, Ward linkage and Euclidean distance achieved the highest silhouette score of 0.95. These parameter choices ensured effective cluster separation and classification of benign and malicious flows.

The distribution of reputation scores among the three source IP addresses identified during DDoS detection experiments is shown in Fig. 7. The chart provides a visual comparison of how individual IP addresses are assessed by the reputation-based detection framework, which is a central component of the proposed mitigation strategy.

In this analysis, each color-coded bar represents the reputation score assigned to a specific source IP, reflecting its observed behavior within the network during both normal and attack scenarios. The green bar for IP 10.0.0.2 indicates a reputation score of 2.61, which is relatively benign based on its traffic characteristics and the probability of malicious activity derived from model analysis. On the contrary, the red bars for IPs 10.0.0.3 and 10.0.0.5 correspond to substantially higher reputation scores of 9.75 and 9.99, respectively.

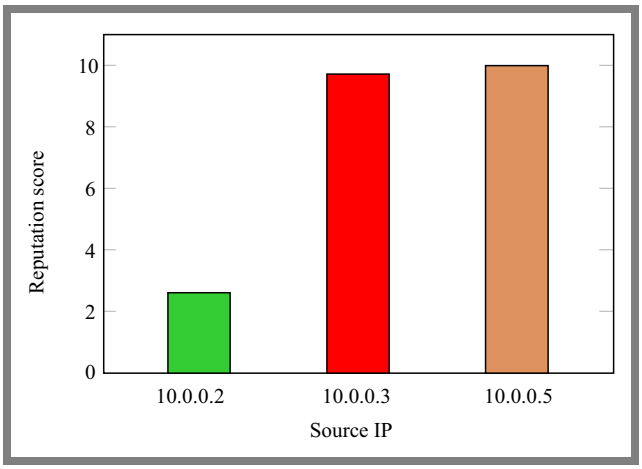

Fig. 7. Source IPs by reputation score.

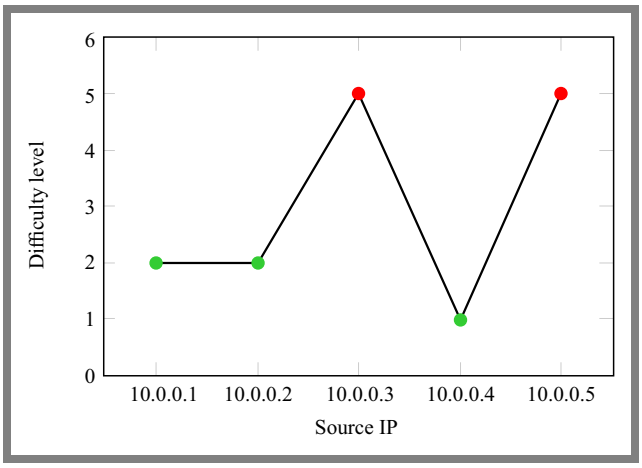

Fig. 8. Difficulty level of source IPs.

The evaluated difficulty levels assigned to the top five source IP addresses observed during the DDoS mitigation experiment in the emulated SDN environment are presented in Fig. 8. Each point on the graph represents a unique source IP, with its corresponding difficulty score, as determined by the game theory PoW mechanism integrated within the mitigation framework.

The plotted line connects the five IP addresses (10.0.0.1, 10.0.0.2, 10.0.0.3, 10.0.0.4, and 10.0.0.5), facilitating visual comparison of difficulty allocation between sources. The difficulty score reflects the level of challenge imposed on each source when attempting to transmit further traffic, with higher values denoting a greater degree of suspicion of malicious intent.

To distinguish between benign and potentially malicious traffic, the data points are color coded: green dots correspond to source IPs 10.0.0.1, 10.0.0.2, and 10.0.0.4, which are classified as benign, while red dots indicate sources that have been assigned high difficulty scores (10.0.0.3, 10.0.0.5) based on their behavior and are identified as malicious. This differentiation is derived from the Bayesian reputation score and the probability of malicious activity, as outlined in the methodology. When the calculated probability that an IP is malicious exceeds a specified threshold, the system responds by increas-

Tab. 10. Analysis of the execution time of proposed methodology.

Stage	Description	Average time [s]
Feature selection (one-time)	Identification of the top 10 features from 80+ attributes	12 – 15
Clustering	Agglomerative clustering of traffic flows	1.5 – 2.0
Reputation scoring	Bayesian probability and reputation computation	0.8 – 1.2
Puzzle assignment and mitigation	Difficulty allocation and enforcement at controller	0.7 – 1.3
Total per monitoring cycle	The periodic cycle includes clustering, reputation scoring, and mitigation, while the one-time feature selection phase is excluded	3 – 5

ing the difficulty level, thus slowing or restricting the actions of these suspicious hosts.

The results discuss patterns about how the detection and mitigation system works in practice. The high values seen in the silhouette and the CH index show that once the system processes and selects key features from the network data, it is able to group normal and attack traffic with clear boundaries. This means that the approach is successful not just with one kind of data, but across different types of network environments, including complex IoT setups and simulated SDN traffic, where network behavior is often unpredictable.

One key insight is that the agglomerative clustering method consistently forms well-separated groups, even in datasets where attacks might be less comparable to normal traffic. This is important because, in real-world networks, attack patterns change constantly, and normal traffic can take many forms. The system's performance suggests that it can keep up with these changes, making it a flexible option for real deployments. In situations where other models, such as K-means or isolation forest, do not distinguish normal and malicious activity, this approach continues to find reliable clusters.

4.3. Processing Time Analysis

The computational efficiency of the proposed methodology was evaluated alongside the clustering performance, measured using the silhouette score represented in Tab. 10.

The feature selection phase, executed once during initialization, required approximately 12 to 15 s when applied to data sets with more than 80 flow-level attributes. From this pool, the proposed methodology consistently retained only 10 highly discriminative features, reducing dimensionality, and improving efficiency. As feature selection is a one-time process, it is excluded from periodic execution, thus eliminating redundant overhead. Each monitoring cycle, comprising clustering, reputation scoring, and PoW puzzle assignment, was completed in 3 to 5 s. The reduced feature set further improves scalability, and future optimization through parallelization or GPU acceleration could enhance performance in larger topologies with heavy traffic.

The authors of [3] used a whale optimization algorithm-based clustering (WOA-DD) for the detection of DDoS in SDN, achieving moderate adaptability but reporting higher false positives, with cluster compactness metrics not exceeding a silhouette score of 0.70 in comparable scenarios. In [4], an entropy-based method is used that achieved detection rate improvements of 6.25 to 20.26% for high-rate attacks, but suffered a notable drop in accuracy for low-rate attacks, with false positive reductions limited to a level between 64.81 and 77.54%. The authors of [5] applied K-means clustering in a semi-supervised setting, which yielded faster classification but lower clustering cohesion, with silhouette scores around 0.60 and DB index values above 0.78. In [6] CAPoW, a context-sensitive AI-assisted PoW framework was developed, reaching a classification accuracy level of 96%, but without unsupervised anomaly detection, which resulted in no silhouette or Davies–Bouldin benchmarks for heterogeneous datasets. The authors of [8] proposed a dynamic game-theoretic defense in SDN that reduced attack traffic by more than 90%, but lacked integrated multi-feature selection and clustering for early detection.

On the contrary, the proposed hybrid approach achieved silhouette scores of 0.8658 for InSDN and 0.9558 for Mininet, with low DB index values of 0.0951 and 0.0392, and high CH index scores exceeding 56 000 for InSDN and 13 900 for Mininet, ensuring clearer separation between attack and legitimate traffic with minimal false positives.

5. Conclusions

The proposed framework integrates multimethod feature selection, unsupervised anomaly detection, and adaptive game-theoretic mitigation to protect against DDoS attacks in SDN environments. Evaluations performed using CICDDoS 2019, InSDN, CICIoT 2023, and Mininet emulation confirmed its effectiveness, with agglomerative clustering achieving low DB index values of 0.0951 for InSDN and 0.0392 for Mininet, together with high CH scores that indicate clear separation between legitimate and malicious traffic.

The adaptive PoW mechanism, guided by posterior probability using reputation scores, ensured that only malicious IP sources obtained a high reputation score of 9.99 with IP 10.0.0.5 that failed the puzzle threshold and triggered targeted defense, while benign hosts, like IP 10.0.0.2 with a score of 2.61, experienced minimal challenge and offered uninterrupted service. This selective, multilayered approach achieved minimal false positives, adapted effectively to mitigate malicious traffic, and demonstrated strong readiness for deployment in practical programmable networks, with the game-theoretic model providing one of the most effective strategies for allocating defense resources efficiently while keeping legitimate network activities unaffected.

References

[1] A.K. Jain, H. Shukla, and D. Goel, "A Comprehensive Survey on DDoS Detection, Mitigation, and Defense Strategies in Software-defined Networks", *Cluster Computing*, vol. 27, pp. 13129–13164, 2024 (https://doi.org/10.1007/s10586-024-04596-z).

[2] A.A. Bahashwan *et al.*, "A Systematic Literature Review on Machine Learning and Deep Learning Approaches for Detecting DDoS Attacks in Software-defined Networking", *Sensors*, vol. 23, art. no. 4441, 2023 (https://doi.org/10.3390/s23094441).

[3] M. Shakil *et al.*, "A Novel Dynamic Framework to Detect DDoS in SDN Using Metaheuristic Clustering", *Transactions on Emerging Telecommunications Technologies*, vol. 33, art. no. e3622, 2022 (https://doi.org/10.1002/ett.3622).

[4] M.A. Aladaileh *et al.*, "Effectiveness of an Entropy-based Approach for Detecting Low- and High-rate DDoS Attacks Against the SDN Controller: Experimental Analysis", *Applied Sciences*, vol. 13, art. no. 775, 2023 (https://doi.org/10.3390/app13020775).

[5] M.N. Jasim and M.T. Gaata, "K-means Clustering-based Semi-supervised for DDoS Attacks Classification", *Bulletin of Electrical Engineering and Informatics*, vol. 11, pp. 3570–3576, 2022 (https://doi.org/10.11591/eei.v11i6.4353).

[6] T. Chakraborty, S. Mitra, and S. Mittal, "CAPoW: Context-aware AI-assisted Proof of Work Based DDoS Defense", *Proc. of the 20th International Conference on Security and Cryptography (SECRYPT)*, vol. 1, pp. 62–72, 2023 (https://doi.org/10.5220/0012069000003555).

[7] T. Chakraborty, S. Mitra, and S. Mittal, and M. Young, "AI_Adaptive_POW: An AI Assisted Proof of Work (POW) Framework for DDoS Defense", *Software Impacts*, vol. 13, art. no. 100335, 2022 (https://doi.org/10.1016/j.simpa.2022.100335).

[8] A. Chowdhary, S. Pisharody, A. Alshamrani, and D. Huang, "Dynamic Game-based Security Framework in SDN-enabled Cloud Networking Environments", *Proc. of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*, pp. 53–58, 2017 (https://doi.org/10.1145/3040992.3040998).

[9] Y. Zhou *et al.*, "Cost-effective Dynamic Shuffling for Mitigating DDoS Attacks Using Moving Target Defense", *Proc. of 6th ACM Workshop on Moving Target Defense (MTD'19)*, pp. 57–66, 2019 (https://doi.org/10.1145/3338468.3356824).

[10] M.V.O. De Assis, A.H. Hamamoto, T. Abrão, and M.L. Proença Jr., "A Game Theoretical Based System Using Holt-winters and Genetic Algorithm With Fuzzy Logic for DoS/DDoS Mitigation on SDN Networks", *IEEE Access*, vol. 5, pp. 9485–9496, 2017 (https://doi.org/10.1109/ACCESS.2017.2702341).

[11] Q. He *et al.*, "A Game-theoretical Approach for Mitigating Edge DDoS Attacks", *IEEE Transactions on Dependable and Secure Computing*, vol. 19, pp. 2333–2348, 2022 (https://doi.org/10.1109/TDSC.2021.3055559).

[12] M. Priyadarsini, P. Bera, S.K. Das, and M.A. Rahman, "A Security Enforcement Framework for SDN Controller Using Game Theoretic Approach", *IEEE Transactions on Dependable and Secure Computing*, vol. 20, pp. 1500–1515, 2023 (https://doi.org/10.1109/TDSC.2022.3158690).

[13] P. Gulihar and B.B. Gupta, "Anomaly-based Mitigation of Volumetric DDoS Attack Using Client Puzzle as Proof-of-Work", *3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, Bangalore, India, 2018 (https://doi.org/10.1109/RTEICT42901.2018.9012127).

[14] E. Okewu, S. Misra, U. Diala, and E.B. Fernandez, "Anti-DDoS Firewall: A Zero-sum Mitigation Game Model for Distributed Denial of Service Attack Using Linear Programming", *4th IEEE International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, Iran, 2017 (https://doi.org/10.1109/KBEI.2017.8324973).

[15] C. Guo, S. Wang, X. Rong, and X. Tao, "Game-theoretic Modeling of Hybrid Defense Strategies Against DRDoS Traffic in 5G Networks", *IEEE International Conference on Communications (ICC)*, Denver, USA, 2024 (https://doi.org/10.1109/ICC51166.2024.10622381).

[16] K.-Y. Sung and S.-W. Hsiao, "Mitigating DDoS with PoW and Game Theory", *2019 IEEE International Conference on Big Data (Big Data)*, Los Angeles, USA, 2019 (https://doi.org/10.1109/BigData47090.2019.9006081).

[17] P. Cotae and R. Rabie, "On a Game Theoretic Approach to Detect the Low-rate Denial of Service Attacks", *2018 International Conference on Communications (COMM)*, Bucharest, Romania, 2019 (https://doi.org/10.1109/ICComm.2018.8429980).

[18] Z. Li, B. Yang, X. Zhang, and C. Guo, "DDoS Defense Method in Software-defined Space-air-ground Network from Dynamic Bayesian Game Perspective", *Security and Communication Networks*, vol. 2022, art. no. 1886516, 2022 (https://doi.org/10.1155/2022/1886516).

[19] I. Sharafaldin, A.H. Lashkari, and A.A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", *Proc. of the International Conference on Information Systems Security and Privacy (ICISSP)*, pp. 108–116, 2019 (https://doi.org/10.5220/0006639801080116).

[20] M.S. Elsayed, N.-A. Le-Khac, and A.D. Jurcut, "InSDN: A Novel SDN Intrusion Dataset", *IEEE Access*, vol. 8, pp. 165263–165284 2020, (https://doi.org/10.1109/ACCESS.2020.3022633).

[21] E.C.P. Neto *et al.*, "CICIoT2023: A Real-time Dataset and Benchmark for Large-scale Attacks in IoT Environment", *Sensors*, vol. 23, art. no. 5941, 2023 (https://doi.org/10.3390/s23135941).

Amit Kachavimath, M.Tech.
School of Computer Science and Engineering
https://orcid.org/0000-0002-8917-8678
E-mail: amitk@kletech.ac.in
KLE Technological University, Hubballi, Karnataka, India
https://www.kletech.ac.in

Narayan D.G., Ph.D.
School of Computer Science and Engineering
https://orcid.org/0000-0002-2843-8931
E-mail: narayan_dg@kletech.ac.in
KLE Technological University, Hubballi, Karnataka, India
https://www.kletech.ac.in

Virtual Machine Placement in Cloud Environments Using a Hybrid Cuckoo Search and Bat Algorithm

Sifeddine Benflis, Sonia-Sabrina Bendib, Sedrati Maamar, Fatima Z. Cherhabil, and Hanane Merouani

University of Batna 2, Batna, Algeria

https://doi.org/10.26636/jtit.2025.4.2244

Abstract — The growing popularity of on-demand pay-as-you-go subscription models for online cloud computing requires increasing amounts of resources to ensure adequate quality of services. However, to satisfy the strong demand for these services, cloud infrastructure providers continue to scale up their data centers. This scaling often lacks an optimal resource management approach, thus leading to inefficiencies, excessive energy consumption, and higher costs. This creates challenges in the virtual machine placement (VMP) process focusing on identifying efficient ways for assigning virtual machines to physical hardware. This paper introduces a hybrid cuckoo search bat algorithm (HCS-BA) to solve VMP in heterogeneous cloud environments. The suitability of the cuckoo search algorithm for global searches is combined with the local refining capacity of the bat algorithm, therefore optimizing both energy consumption and resource utilization. The results of simulations carried out in Matlab and CloudSim for scalability testing demonstrate that HCS-BA outperforms both individual algorithms. It reduces energy consumption and improves resource utilization.

Keywords — bat algorithm, cloud computing, cuckoo search algorithm, virtual machine placement

1. Introduction

Cloud computing is an evolving technology that offers, over the Internet, a vast majority of services to a large number of clients. These services can be obtained on demand, without any interaction with the cloud provider, via the pay-as-you-go model which turns computing resources into business assets and allows to charge consumers for the capacity they have used.

As cloud services continue to evolve, the demand for larger and more efficient data centers is booming. These facilities require massive computing, networking, and data storage capacities to support various applications with the expected quality of services. Consequently, the growing reliance on cloud environments is driving the construction of new, more advanced data centers. However, this rapid expansion introduces significant challenges, including high energy consumption, resource usage inefficiencies, and network bandwidth limitations [1]–[3].

Virtualization has become the cornerstone of cloud computing, allowing physical resources to be efficiently shared among multiple users through virtual machines (VMs). These virtual machines run on physical hardware housed within data centers, leading to significant improvements in resource utilization [4]. However, this advancement also creates new challenges, most notably the virtual machine placement (VMP) problem. Therefore, addressing VMP has become essential for enhancing the performance and efficiency of cloud infrastructure.

The VMP problem involves mapping VMs to physical machines (PMs) within a cloud data center. It is a key consideration in how these data centers operate and plays a crucial role in cloud computing, as it significantly affects performance and costs. Placing virtual machines without optimization can result in significant inefficiencies such as underutilized servers, increased energy consumption from running unnecessary PMs, and idle servers that waste power.

As cloud services continue to grow and demand for computing resources increases, VMP has become a major area of interest for both researchers and practitioners. This challenge has been studied from various perspectives, with many approaches aiming to minimize energy use and improve overall system efficiency. Despite the often conflicting objectives, the diversity in strategies has led to a wide range of models, frameworks, and algorithms [5], [6].

Given the complexity of the VMP problem and its impact on efficiency, researchers have turned to heuristic and metaheuristic algorithms to find near-optimal solutions. Various approaches have been widely adopted, including ant colony optimization [2], whale optimization [7], cuckoo search algorithm (CSA) [8], and bat algorithm (BA) [9]. Furthermore, researchers have combined techniques such as hybridization of flower pollination with particle swarm optimization to take advantage of the features of multiple algorithms [4].

Over the years, both BA and CSA have been applied to a wide range of optimization problems, including VMP. The cuckoo search algorithm is known for its strong exploration capabilities, thanks to the Lévy flight mechanism [10], while BA effectively balances exploration and exploitation using adaptive parameters such as loudness and pulse rate [11]. These strengths have made both algorithms popular in the face of complex challenges. However, despite their proven

potential, there has been relatively little research focused on combining these two algorithms for VMP-related tasks. While each excels in different aspects (CSA in broad exploration and BA in fine-tuned exploitation), their hybridization remains largely unexplored.

Merging their complementary features could offer a promising solution to common issues, for instance premature convergence or getting caught in local optima. Exploring this hybrid approach could open new avenues to improve the efficiency and effectiveness of VMP in cloud environments.

In this paper, a hybrid algorithm has been proposed balancing the exploitation capabilities of BA with the exploration aptitude of the Lévy flight-relying CS algorithm (HCS-BA). This hybrid method is designed to optimize VMP in cloud computing environments while simultaneously reducing energy consumption and improving resource utilization.

The rest of this paper is organized as follows. Section 2 discusses related work. Problem formulation is introduced in Section 3 and the HCS-BA algorithm is described in Section 4. Section 5 presents and discusses the experimental results. The paper is concluded in Section 6.

2. Related Work

The process of selecting a VM that should be assigned to given physical machine (PM) is known as virtual machine placement. This task becomes challenging due to factors such as the scale and complexity of the cloud environment, which bring concerns such as efficient resource utilization, energy consumption, and overall system performance. To address these challenges, researchers have proposed a wide range of techniques, ranging from heuristic and metaheuristic methods to hybrid approaches, all aiming to find smarter and more efficient placement strategies.

Existing methods, such as flower pollination optimization (FPO) and particle swarm optimization (PSO), often struggle with local optima and result in inefficient resource utilization. To overcome the drawback of both algorithms, a novel multi-objective algorithm has been developed, known as hybrid particle swarm with Lévy flight flower pollination optimization (HPSOLF-FPO), being a hybrid of PSO with Levy flight and flower pollination, to minimize the time needed for the mapping of VM to PM, energy consumption, and resource waste [4]. It takes advantage of the exploitation strength of PSO and exploration strength of FPO, resulting in an improvement in all performance metrics. Unfortunately, HPSOLF-FPO still suffers from some limitations, mainly the congestion problem, meaning that too many VMs are allocated to a single PM. Local optima pose a challenge too, as they cannot be solved even with the help of Lévy flight.

In [12], an innovative method to address the high energy demands of cloud data centers has been introduced, i.e. a hybrid VMP algorithm that integrates a genetic algorithm for permutation-based optimization problems (IGA-POP) with a multidimensional resource-aware best fit (BF) allocation strategy. This combination reduces the number of active physical servers, leading to significant energy savings, and simultaneously reduced resource waste. IGA-POP balances exploration and exploitation within the optimization process, while BF strategy ensures the efficient use of critical resources such as CPU, RAM, and network bandwidth. The experimental results highlight its superiority in terms of energy efficiency and resource utilization, compared to traditional heuristic and metaheuristic approaches.

The problem of reducing energy consumption for dynamic virtual machine placement in cloud data centers has been tackled in [13]. The authors proposed an ant colony system (ACS) that was enhanced with designed heuristics. This approach dynamically adapts to workload fluctuations and prioritizes utilizing active PMs over activating new ones, thus minimizing total energy use. Improvements in precision and efficiency have been observed when a problem is defined as a constrained combinatorial optimization problem. Extensive simulations demonstrate that the proposed ACS outperforms traditional methods, such as first-fit decreasing (FFD) and existing ACS-based strategies across small- to large-scale data centers.

Paper [8] introduced an advanced optimization technique for efficient allocation of virtual physical machines. By improving the cuckoo search algorithm (CSA) from [14], with cost and perturbation functions, the study addresses resource utilization, reducing the number of active PMs and energy usage. The proposed approach ensures minimal resource waste while avoiding server overload. Experiments on benchmark datasets reveal its superiority to traditional methods, as it achieves reduced energy consumption and fewer active servers while maintaining fast task execution.

In the context of hybridization, the method combining flower pollination optimization (FPO) and the Pareto front module of the nondominated classification genetic algorithm (NSGA-II) was used to address the VMP problem in [15]. The proposed FP-NSO algorithm, along with a bio-VMP framework that helps allocate VMs to PMs, optimizes multiple objectives, including maximizing resource utilization and minimizing power consumption, hence reducing carbon emissions in cloud data centers.

Another proposal, the GATA algorithm [16], is a hybrid approach that combines the genetic algorithm (GA) and the tabu search (TS). The optimization objectives focus on reducing energy consumption and improving the balance of load across data center resources. To enhance GA's local search capabilities, TS has been used as a mutation operator, preventing premature convergence and increasing solution diversity. Experimental evaluations show that GATA outperforms the traditional GA, simulated annealing (SA) and ACS-based methods, achieving better energy efficiency, more balanced resource utilization, and competitive execution times.

A VMP method using an enhanced cuckoo search (ECS) was explored in [17]. The ECS integrates strategies such as Lévy flights for local and global exploration, "effective overload detection", "VM selection policies", and a status index for resource utilization. Optimization objectives are minimizing energy consumption, SLA violations, and VM migrations

while maximizing resource utilization. Experiments were conducted using CloudSim, and real workload traces demonstrated the superiority of ECS over other solutions, such as genetic, optimized firefly search (OFS) and ant colony (AC) algorithms, achieving significant energy savings, reduced SLA violations and optimized VM placements. However, even with Lévy flights, CS tends to over-explore, meaning that local optima exploitation may be weak.

An approach called multi-objective bat algorithm with decomposition (MOBA/D), presented in [9], addressed minimizing energy consumption and reducing network traffic. Unlike traditional methods, MOBA/D leverages a decomposition-based strategy to divide the VMP into smaller subproblems, solving them individually for improved efficiency with another technique that allows for the discretization of continuous values, meaning that it lets them deal with the VMP problem more efficiently.

The authors compared the performance of MOBA/D with that of established multiobjective algorithms such as MOEA/D, NSGA-II, and memetic MOEA across various scenarios. Experimental results demonstrate that MOBA/D outperforms these algorithms in terms of Pareto front solutions and execution time.

Despite significant advances in VMP algorithms, existing methods often suffer from limitations such as premature convergence to local optima, inefficient resource utilization, slow convergence speed, unstable exploitation, and weak exploration capabilities.

Study [18] introduced a hybrid approach that combines BA with the CSA to improve task scheduling in cloud settings. In this method, tasks are first ranked using BA and then CS is applied to assign these tasks to virtual machines in a way that balances the workload. The approach was tested in CloudSim and compared with other metaheuristics, showing that HB-CSA achieved better load distribution, improved scheduling efficiency, and more effective VM utilization.

However, the HB-CSA framework was designed specifically for task scheduling and does not address the broader challenge of VMP in heterogeneous cloud data centers. VMP brings additional complexity, as it involves diverse host resources, multidimensional constraints like CPU, memory, and storage, as well as energy consumption considerations. Importantly, heterogeneity and energy optimization were not central to the HB-CSA study, leaving room for approaches that explicitly address these issues. In this context, this paper aims to address these gaps by developing a hybrid algorithm that integrates BA with the CS algorithm for VMP. This combination leverages BA exploitation capabilities with efficient CSA exploration with Lévy flight.

3. Problem Formulation

We consider a cloud environment made up of a set of physical and virtual machines, each with varying resource capacities. The objective is to find the optimal placement of virtual machines in PMs in a way that maximizes resource utilization while minimizing energy consumption.

We have the following sets which define our model:

$PM = \{PM_1, PM_2, \ldots, PM_n\}$: set of physical machines

$VM = \{VM_1, VM_2, \ldots, VM_m\}$: set of virtual machines

$R_{VM_j} = \{\text{CPU}_j, \text{RAM}_j, \text{Storage}_j\}$: resource requirements of VM_j

$C_{PM_i} = \{\text{CPU}_i, \text{RAM}_i, \text{Storage}_i\}$: capacity of physical machine_i

$U_{\text{CPU}} =$ CPU usage of a physical machine

$U_{\text{RAM}} =$ RAM usage of a virtual machine

$U_{\text{Storage}} =$ storage usage of a virtual machine

3.1. Resource Model

The resource model describes how CPU, RAM, and storage usage rates are calculated and utilized in relation to their respective usages.

Resource utilization for each host is defined as:

$$\text{Total RAM usage} = \sum_{j=0}^{m} \text{RAM}_j \,, \tag{1}$$

$$\text{Total CPU usage} = \sum_{j=0}^{m} \text{CPU}_j \,, \tag{2}$$

$$\text{Total storage usage} = \sum_{j=0}^{m} \text{Storage}_j \,, \tag{3}$$

$$U_{\text{CPU}} = \left(\frac{\text{Total CPU usage}}{\text{CPU total capacity}} \right) \times 100 \,, \tag{4}$$

$$U_{\text{RAM}} = \left(\frac{\text{Total RAM usage}}{\text{RAM total capacity}} \right) \times 100 \,, \tag{5}$$

$$U_{\text{Storage}} = \left(\frac{\text{Total storage usage}}{\text{Storage total capacity}} \right) \times 100 \,. \tag{6}$$

The overall utilization across hosts is calculated as the average utilization of CPU, RAM, and storage is:

$$\begin{matrix}\text{Resource} \\ \text{utilization}\end{matrix} = \frac{\sum\limits_{i=0}^{n} U_{\text{CPU}_i} + \sum\limits_{i=0}^{n} U_{\text{Storage}_i} + \sum\limits_{i=0}^{n} U_{\text{RAM}_i}}{3} \,, \tag{7}$$

with the following constraints:

$$\sum_{j=0}^{m} \text{CPU}_j \leqslant C_{\text{PM}_i}^{\text{CPU}} \,, \tag{8}$$

$$\sum_{j=0}^{m} \text{Storage}_j \leqslant C_{\text{PM}_i}^{\text{Storage}} \,, \tag{9}$$

$$\sum_{j=0}^{m} \text{RAM}_j \leqslant C_{\text{PM}_i}^{\text{RAM}} \,. \tag{10}$$

3.2. Energy Model

The energy model estimates the power consumption of each host based on its utilization of resources. For every physical

machine, energy usage is calculated in the following way:

$$E_{\text{host}} = U_{\text{idle}} + (U_{\text{max}} - U_{\text{idle}}) \cdot (\alpha_{\text{cpu}} \cdot U_{\text{cpu}} + \beta_{\text{ram}} \cdot U_{\text{ram}} + \gamma_{\text{storage}} \cdot U_{\text{storage}}) , \quad (11)$$

where: U_{idle} is idle energy consumption, U_{max} is energy consumption in active state, α_{cpu}, β_{ram}, $\gamma_{storage}$ are weights for CPU, RAM, and storage contributions to energy consumption. Total energy consumption can be modeled as follows:

$$\text{Total energy} = \sum_{i=0}^{n} E_{\text{host}_i} . \quad (12)$$

3.3. Fitness Function

Energy consumption and resource utilization can be grouped into one single fitness function:

$$\text{Fitness function} = \alpha \cdot \text{energy} + \frac{1}{\beta \cdot \text{Resource utilization}} , \quad (13)$$

where α – weight for the energy component, β – weight for resource utilization.

4. Proposed Approach

In this investigation, both nests and bats are represented using a matrix **M**, which captures all potential solutions (Fig. 1). In this matrix, rows correspond to PMs and columns correspond to VMs. Each matrix represents a unique way to assign VMs to PMs, effectively illustrating different placement strategies. For each solution, $H \times V$ binary matrix is generated, where the solution $[i, j, k]$ $a \in A[0, 1]$ indicates VM k assigned to host j in the nest or bat i.

4.1. Overview of the Cuckoo Search Algorithm

The cuckoo search algorithm mimics the behavior of cuckoo birds. Cuckoo birds rely on other birds nests to lay their eggs rather than building their own. The host bird might identify the intruding egg, so once that happens, birds either build new nests or simply remove the egg. In a nest, each egg represents a solution, and the cuckoo egg represents a new and good solution. The host bird identifies the cuckoo egg with the probability of Pa, so the higher the probability, the higher the chance of the egg being removed [10], [19].

A Lévy flight is a random walk-in which step-lengths are calculated according to a heavy-tailed probability distribution [10], [19]. This helps prevent premature convergence and being stuck in local optima. This behavior is especially useful in optimization, because it allows the algorithm to explore the search space more efficiently.

M =

	VM1	*VM2*	*VM3*	*VM4*	*VM5*	*VM6*	*VM7*	*VM8*	*VM9*	*VM10*
PM1	1	0	1	0	0	0	1	0	0	0
PM2	0	1	0	0	0	1	0	1	0	0
PM3	0	0	0	1	0	0	0	0	0	0
PM4	0	0	0	0	1	0	0	0	0	0
PM5	0	0	0	1	0	0	0	0	1	1

Fig. 1. Example of VM-PM allocation matrix.

When applied to the VMP problem, each nest can be viewed as a possible allocation of virtual machines to physical hosts. Through the use of Lévy flights, CS is able to make occasional long jumps in the search space, which allows it to explore diverse placement possibilities and avoid being trapped in local optima. Over time, it discovers more effective VM placements by using a P_a probability to determine whether to keep or discard a given solution.

This strong exploratory ability makes the algorithm highly effective in discovering promising global solutions in complex and heterogeneous environments. However, because CS emphasizes global exploration, it often converges more slowly and lacks strong local exploitation capabilities, which means it may not always fine-tune solutions efficiently once good regions of the search space are found.

Algorithm 1 Cuckoo search using Lévy flight

Input: population size N_{pop}, iterations N_{iter}, hosts H, VMs V, abandonment rate P_a, fitness weights α, β, resource constraints.
Output: Best solution S_{best} and its fitness
Generate resource capacities for hosts
Assign random requirements to each VM
Initialize population:
Generate $H \times V$ binary matrix $nests[i, j, k] \in \{0, 1\}$
for each nest **do**
    Randomly assign each VM to one host
end for
Validate VM assignments and adjust overloaded hosts
for $iter = 1$ to N_{iter} **do**
    for each nest **do**
        Evaluate fitness via Eq. (13)
        Update S_{best} if improved
        if $rand > P_a$ **then**
            Apply Lévy flight for exploration
        else
            Replace P_a worst nests with new random ones
        end if
        Validate VM assignments and adjust overloaded hosts
    end for
end for
End

The Algorithm 1 with Lévy flight begins by initializing the problem parameters, such as the number of hosts and VMs, population size (nests), number of iterations, and fitness weights. Each nest represents a possible solution (a binary matrix), where each 1 indicates that a specific VM is placed on a specific host. The initial population of nests is created by randomly assigning VMs to hosts. The algorithm then checks for valid placements and adjusts any overloaded hosts to ensure feasibility. Adjusting overallocated hosts can be achieved by preventing PMs from hosting new VMs which exceed the total capacity of the PM using Eqs. (8)–(10).

During each iteration of the algorithm, the fitness of every candidate solution (or nest) is evaluated on fitness function with energy and resource utilization serving the role of param-

eters. If a solution outperforms the current best, it becomes the new reference point S_{best}. To maintain exploration and avoid premature convergence, the algorithm introduces randomness through two strategies governed by probability P_a.

With a chance higher than P_a, a Lévy flight is applied to explore new and potentially better placements by making large or small shifts in the VM-to-PM assignment. Otherwise, the algorithm abandons the least promising solutions and replaces them with new randomly generated ones to increase diversity in the population. After these updates, the algorithm ensures that all assignments are valid and adjusts any overloaded hosts accordingly. This process repeats over a set number of iterations, gradually refining the solutions, and finally returning the best one found.

4.2. Overview of the Bat Algorithm

Bats rely on a type of sonar called echolocation to figure the distance between them and their prey. They travel in the search space using a set of parameters [11]:

- Velocity: bats randomly move in the search space with V_i.
- Position: bats move from one position to another, represented by X_i.
- Frequency range: defines the lower and upper bounds of frequency and controls the range of velocity adjustments for bats.
- Loudness: represents the initial loudness of bats, controls step size and convergence behavior, and typically decreases over iterations as bats approach their prey.
- Pulse emission rate: defines the initial rate of pulse emission and increases as the algorithm progresses, promoting local exploitation.

In the BA applied to VMP, each bat represents a possible solution, essentially a way of assigning virtual machines to physical machines. Bats "fly" through the solution space by adjusting their positions (placements) based on jumps instead of velocity to make the problem discrete.

Instead of moving the bat in certain direction with a specific velocity, we use the *jump* variable which helps us move a VM

$\mathbf{M} =$

	VM1	*VM2*	*VM3*	*VM4*	*VM5*	*VM6*	*VM7*	*VM8*	*VM9*	*VM10*
PM1	1	0	1	0	0	0	1	0	0	0
PM2	0	1	0	0	0	1	0	1	0	0
PM3	0	0	0	1	0	0	0	0	0	0
PM4	0	0	0	0	1	0	0	0	0	0
PM5	0	0	0	1	0	0	0	0	1	1

Jump ↓

$\mathbf{M}' =$

	VM1	*VM2*	*VM3*	*VM4*	*VM5*	*VM6*	*VM7*	*VM8*	*VM9*	*VM10*
PM1	0	0	0	1	0	0	0	0	0	0
PM2	0	0	1	0	1	0	0	0	1	0
PM3	1	1	0	0	0	1	0	1	0	1
PM4	0	0	0	1	0	0	1	0	0	0
PM5	0	0	0	0	0	0	0	0	0	0

Fig. 2. Example of a discrete VM migration jump.

Algorithm 2 Bat algorithm flowchart

Input: population size N_{pop}, max iterations N_{iter}, weights α, β, loudness A, pulse rate r, f_{min}, f_{max}, maxHops, minHops, abandonment rate P_a, and resource constraints
Output: Best solution S_{best} and its fitness
Generate resource capacities for hosts
Assign random requirements to each VM
Initialize population:
Generate $H \times V$ binary matrix $bats[i, j, k] \in \{0, 1\}$
for each bat **do**
    Randomly assign each VM to one host
end for
Validate VM assignments and adjust overloaded hosts
for $iter = 1$ to N_{iter} **do**
    for each bat **do**
        Generate frequency $f \in [f_{min}, f_{max}]$
        Update VM assignments via frequency-guided jumps
        Evaluate fitness using Eq. (13)
        if fitness improved **then**
            Update global best solution
        end if
        if $rand > r$ **then**
            Perform local random walk around best solution
            Fine-tune using A-scaled adjustment
        end if
        if $rand < A$ **then**
            Decrease A and increase r
        end if
        Validate VM assignments and adjust overloaded hosts
    end for
end for
End

from one PM to another. Each bat is considered a candidate solution to the problem. The *jump* is calculated as follows:

$$jump = minHops + \text{random}(maxHops - minHops + 1)\,, \tag{14}$$

where $minHops$ represents the minimum number of hops, $maxHops$ represents the maximum number of hops, random picks a number between 0 and $(maxHops - minHops + 1)$. The bat algorithm also uses frequency, loudness, and pulse rate to control exploration and exploitation. These parameters control how much the bat explores new solutions or fine-tunes existing ones. At each step, a bat can explore globally or exploit locally, depending on how good the current solution is and how close it is to the best one found so far. Over time, bats naturally focus more on promising areas. The best solution discovered during this process is returned as the optimal placement.

Overall, BA is particularly effective in fine-tuning solutions. Its adaptive use of loudness and pulse emission rate allows it to gradually shift focus toward promising regions of the search space, making it well suited for refining virtual machine place-

ments once good candidates are found. Unfortunately, this strength comes with a limitation. Because the algorithm emphasizes local search, it can sometimes converge too quickly and miss better global solutions. In large and heterogeneous cloud environments, this tendency toward premature convergence highlights the need to complement the BA with stronger exploratory capabilities.

The BA shown as Algorithm 2 starts by setting up the cloud environment defining the resource capacities of each physical host and the demands of each virtual machine. Each bat in the population represents a possible way to place VMs onto hosts, initialized randomly using a binary matrix. VM placement is validated and overallocation is adjusted by preventing PMs from hosting new VMs that exceed the total capacity of the PM using the constraints (8)–(10). Then, the algorithm enters its main loop. Here, each bat updates its position using a frequency value, which guides how it shifts VM assignments.

Each updated solution is evaluated using fitness functions that consider energy efficiency and resource utilization. If a bat discovers a better solution, it becomes the new global best. Occasionally, if a random chance exceeds the bat's pulse rate, it performs a local search near the current best solution, fine-tuning it slightly based on its loudness. Then, with a probability based on A, the algorithm decides whether to accept the new solution. If accepted, it simulates a bat coming closer to its prey by decreasing the volume and increasing pulse rate, which gradually shifts the bat's behavior toward exploitation. Finally, it checks and adjusts the VM assignments to ensure that no host is overloaded.

4.3. Hybrid Cuckoo Search-Bat Algorithm (HCS-BA)

The major strength of the proposed hybrid cuckoo search-bat algorithm (HCS-BA) shown as Algorithm 3, is not just the combination of two metaheuristics, but the way in which this integration is structured to address the specific challenges of VMP in heterogeneous cloud environments.

While hybrid metaheuristics are common in the literature, most of them follow a sequential or operator-level fusion that often inherits the weaknesses of both methods. In contrast, HCS-BA employs a competitive, parallel design that preserves the independent search behavior of each algorithm and leverages their complementary strengths through a best-solution selection strategy. This ensures that the algorithm avoids premature convergence through explorations, while also accelerating refinement once promising solutions emerge through exploitation.

HCS-BA starts by generating the resource capacities (CPU, RAM, storage) for each physical host and assigning random requirements to the virtual machines. Two separate populations are initialized: one for CS (nests) and one for BA (bats), with each individual represented as a binary matrix indicating VM-to-host assignments.

The main loop runs for a set number of iterations. In the CS phase, the solution of each nest is evaluated using the fitness function. If a solution outperforms the current best, it becomes the new reference point S_{best} for nests. With probability P_a, the algorithm either applies a Lévy flight to explore new

Algorithm 3 HCS-BA scheme

Input: population size N_{pop}, max iterations N_{iter}, weights α, β, loudness A, pulse rate r, f_{min}, f_{max}, maxHops, minHops, abandonment rate P_a, and resource constraints
Output: Best solution S_{best} and its fitness
Generate host capacities (RAM, storage, CPU)
Assign random resource requirements to each VM
Initialize population:
Generate two $H \times V$ binary matrices:
    $bats[i,j,k], nests[i,j,k] \in \{0,1\}$
    (VM k assigned to host j)
for each nest and bat **do**
    Randomly assign each VM to one host
end for
Validate VM assignments and adjust overloaded
for $t = 1$ to N_{iter} **do**
                                ▷ cuckoo search phase
    for each nest **do**
        Evaluate fitness via Eq. (13)
        Update S_{best} if improved
        if $rand > P_a$ **then**
            Apply Lévy flight for exploration
        else
            Replace P_a worst nests with new random ones
        end if
        Validate VM assignments and adjust overloaded hosts
    end for
                                ▷ bat algorithm phase
    for each bat **do**
        Generate frequency $f \in [f_{min}, f_{max}]$
        Update VM assignments via frequency-guided jumps
        Evaluate fitness using Eq. (13)
        if fitness improved **then**
            Update global best solution
        end if
        if $rand > r$ **then**
            Perform local random walk around best solution
            Fine-tune using A-scaled adjustment
        end if
        if $rand < A$ **then**
            Decrease A and increase r
        end if
        Validate VM assignments and adjust overloaded hosts
    end for
                                ▷ Fitness comparison
    Compare best fitness of bats and nests
    Keep best individual for next iteration
end for
End

placements or replaces the worst-performing nests with fresh solutions. All placements are then validated to avoid host overloads.

Next comes the BA phase, where each bat adjusts its solution using frequency-guided jumps. If a solution outperforms the current best, it becomes the new reference point S_{best} for bats. Bats may also perform a local random walk around the best-known solution, fine-tuning it using their loudness parameter A. Over time, the noise decreases while pulse rate r increases, gradually shifting their behavior from exploration to exploitation. VM placements are re-validated after each update.

Finally, in the fitness comparison phase, the best solutions from both algorithms are compared. The one with the best fitness is retained for the next iteration, ensuring that the algorithm progresses by leveraging the strengths of both exploration (cuckoo) and exploitation (bat). This collaborative approach aims to balance diversity and refinement, guiding the search toward optimal placements of virtual machines.

5. Results and Experiments

To evaluate the proposed algorithm, we performed a series of simulations using both Matlab and CloudSim. Matlab experiments focused on implementing the core optimization logic and comparing the proposed hybrid approach with several well-known metaheuristic algorithms: bat algorithm (BA), cuckoo search algorithm (CSA), ant colony optimization (ACO), particle swarm optimization (PSO), and non-dominated sorting genetic algorithm (NSGA-II). Each experiment was carried out on a set-up consisting of 20 physical hosts, while the number of virtual machines was varied over four scenarios between 20, 50, 100, and 200.

To further test the scalability of the approach, we conducted additional experiments using CloudSim, in collaboration with the (HPC) laboratory at the University of Trento. Their cluster environment provided the computational power needed to simulate large-scale cloud data centers and compare the HCS-BA algorithm with its standalone variants (BA and CSA) under realistic load conditions. The experimental setup also incorporated heterogeneity at the VM level. Although all hosts shared identical configurations (each with 64 GB of RAM, 100 GB of storage, and 10 000 MIPS), the VMs were intentionally configured with CPU capacities of (1 000 MIPS), RAM (2 to 4 GB) and storage (5 to 10 GB).

The proposed HCS-BA algorithm handles heterogeneity naturally by evaluating the quality of each solution. Since every virtual machine has different resource requirements, the fitness function checks how well these demands are met by the available capacities of each host.

To assess the effectiveness of the HCS-BA algorithm, we measured the following key performance indicators:

- Energy consumption – the total energy usage of active hosts after VM placement,

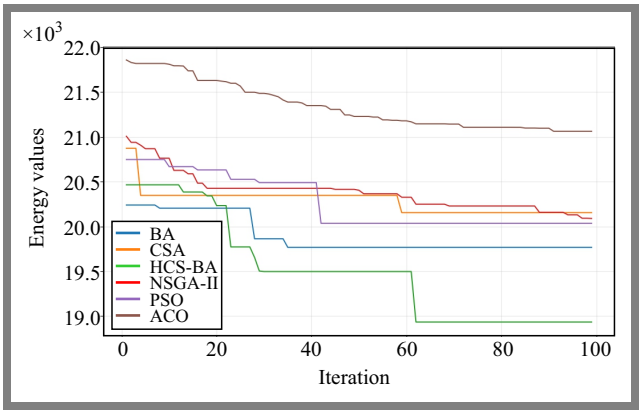

Fig. 3. Energy consumption over iterations for the 200 VM set.

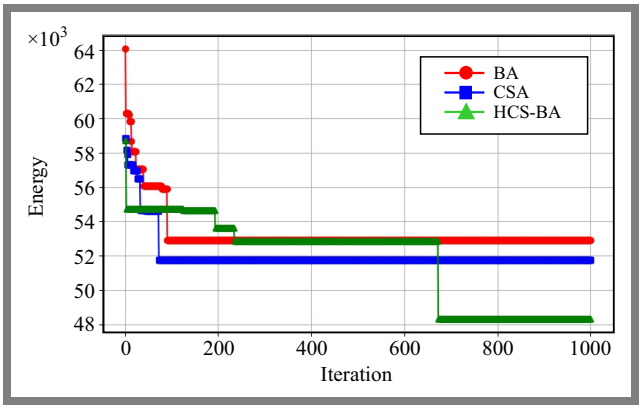

Fig. 4. Energy consumption over iterations for 1000 VMs.

- Resource utilization, the percentage of CPU, RAM, and storage utilization across all hosts,
- Fitness score – the optimized value that balances energy efficiency and resource utilization.

5.1. Energy Consumption

As illustrated in the results, the HCS-BA algorithm consistently outperforms all other compared methods, including CSA, BA, ACO, PSO, and NSGA-II, by achieving significantly lower energy consumption and faster convergence rates. Figure 3 highlights how HCS-BA rapidly stabilizes at an optimal energy level. Meanwhile, the CloudSim-based scalability test presented in Fig. 4 further confirms the robustness of the approach even as the system scales, since HCS-BA maintains its advantage over the standalone CSA and BA, proving both its adaptability and effectiveness in large-scale cloud environments.

5.2. Energy Consumption over Different VMs

As shown in Fig. 5, the HCS-BA algorithm provides strong and consistent performance across all VM set sizes. Although ACO slightly outperforms it in smaller configurations (20, 50, and 100 VMs), HCS-BA remains highly competitive, maintaining low energy consumption levels. However, as the environment scales to 200 VMs, HCS-BA clearly takes the lead, outperforming all other algorithms including CSA, BA, PSO, and NSGA-II. This demonstrates that the hybrid

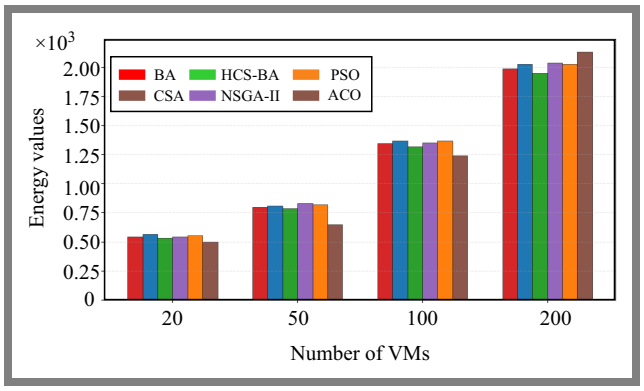

Fig. 5. Energy consumption on different VM set.

approach becomes increasingly effective as the system grows, showcasing its ability to efficiently handle larger and more complex cloud workloads.

5.3. Fitness Score

The fitness function analysis shown in Fig. 6 highlights how each algorithm converges over 100 iterations. CSA shows a quick initial drop in fitness values, but soon levels off at a higher point, suggesting that it settles for less optimal solutions due to its stronger focus on global exploration. In contrast, BA improves more steadily, showing good local search ability but slower overall convergence.

The hybrid HCS-BA, however, combines the strengths of both achieving the lowest fitness values by approximately iteration 60 and converging much faster. This demonstrates its effective balance between exploration and exploitation, leading to more accurate and energy-efficient virtual machine placement in cloud environments. Overall, HCS-BA not only outperforms standalone CSA and BA, but also surpasses other approaches like PSO, ACO, and NSGA-II, demonstrating its robustness, scalability, and superior optimization capability in large-scale scenarios.

Figure 7 presents the fitness values obtained from the CloudSim simulations used to evaluate the scalability of the algorithms. It clearly shows that the HCS-BA algorithm achieves lower fitness values compared to the standalone BA and CSA approaches, demonstrating its superior optimization capability. As the iterations progress, the HCS-BA converges more efficiently, maintaining stability even as the problem scale increases. This confirms that HCS-BA not only scales effectively in larger cloud environments, but also consistently delivers better fitness results.

Figure 7 also shows how well the algorithms scale. It is clear that the HCS-BA algorithm performs better, reaching better fitness values than BA and CSA. As the iterations progress, HCS-BA converges more smoothly and efficiently, showing that it can handle larger and more complex workloads without losing performance.

5.4. Fitness Score over Different VMs

As shown in Fig. 8, the HCS-BA algorithm maintains strong performance across all VM set sizes, achieving lower fitness

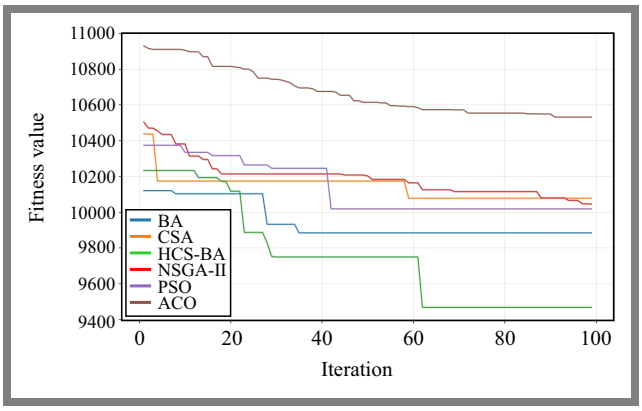

Fig. 6. Fitness score over iterations for a 200 VM set.

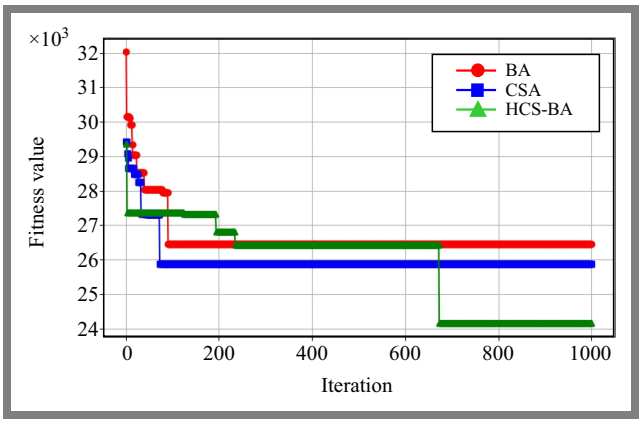

Fig. 7. Fitness score over iterations for a 1000 VM set.

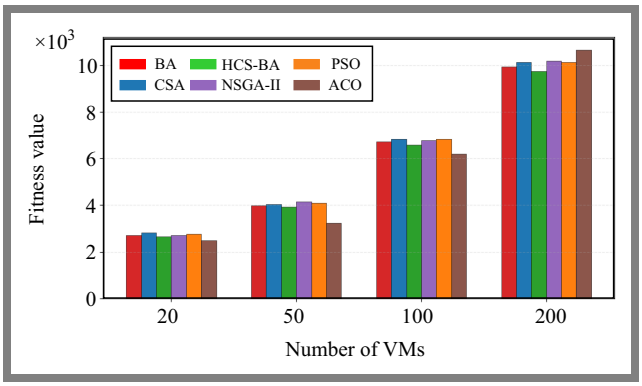

Fig. 8. Fitness score over different sizes of VM set size.

scores that reflect its efficient optimization behavior. Although ACO performs slightly better in smaller configurations (20, 50, and 100 VMs), HCS-BA remains highly competitive and stable. However, as the number of VMs increases to 200, HCSBA clearly stands out, outperforming all other algorithms, including CSA, BA, PSO, and NSGA-II. This demonstrates the impressive scalability and ability to deliver consistent high-quality results even as the complexity of the cloud environment increases.

5.5. Resource Utilization

Figure 9, generated using Matlab, compares resource utilization across different algorithms. Among them, NSGA-II achieves the highest resource utilization, outperforming all

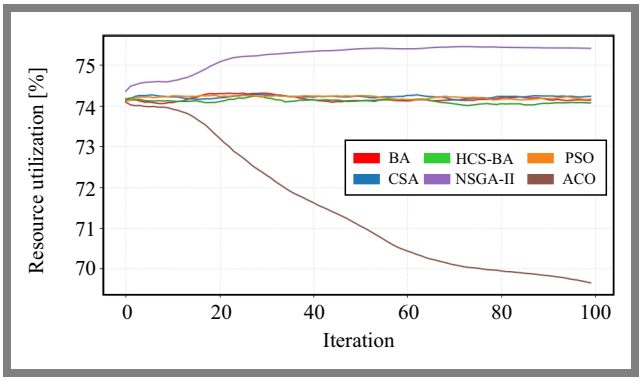

Fig. 9. Resource utilization over iterations for a 200 VM set.

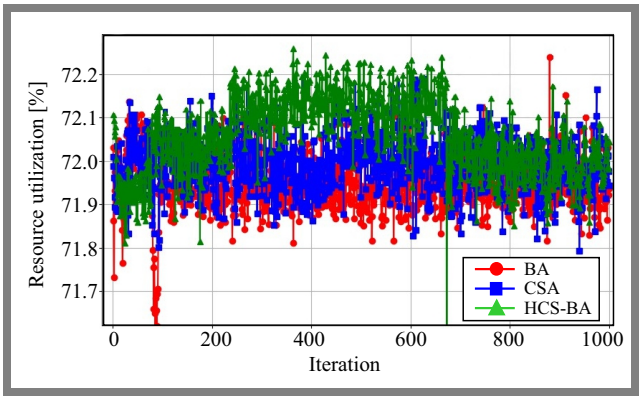

Fig. 10. Resource utilization over iterations for a 1000 VM set size.

other algorithms. HCS-BA, however, maintains a strong and stable performance, closely following PSO and its standalone variants (BA and CSA). On the other hand, ACO records the lowest utilization, indicating weaker efficiency in managing host resources. The CloudSim output presented in Fig. 10 shows that HCS-BA consistently achieves the highest resource utilization and remains stable throughout the simulation, outperforming both the standalone BA and CSA. This consistency demonstrates that HCS-BA scales effectively.

Several previous studies have shown notable energy savings. For example, in paper [13], a medium-scale data center achieved an energy improvement of 4.92%, while a large-scale one achieved 3.69%. Similarly, in [20], a reduction in energy consumption of 7% and 12% was reported using the non-dominated vector generation (ONVG) and spacing methods, respectively.

Another work [21] reported decreases of 28%, 27%, 24%, and 1% for four different algorithms. Although the mean resource utilization in some of these studies was slightly lower than the best performing baselines, they still outperformed many comparative methods.

In this work, the proposed HCS-BA shows that it consistently outperforms all other approaches in terms of energy efficiency. Compared to BA, CSA, NSGA-II, PSO, and ACO, HCS-BA achieves energy reductions of approximately 1.99%, 3.87%, 4.36%, 3.81%, and 8.64%, respectively. It also improves resource utilization by 4% compared to ACO. While HCS-BA shows performance comparable to BA, CSA, and PSO, it struggles when evaluated against NSGAII. Unlike previous studies that focused primarily on energy reduction, the proposed approach emphasizes maintaining high resource utilization efficiency, demonstrating a balanced improvement in both energy efficiency and system performance.

6. Conclusions

The proposed HCS-BA algorithm demonstrates significant improvements in optimizing VMP in heterogeneous cloud environments. Through comprehensive simulations, the hybrid algorithm consistently outperformed standalone CSA and BA and even recent VMP approaches in key performance metrics.

The HCS-BA approach achieved superior results in minimizing energy consumption, maximizing resource utilization, and converged faster to optimal solutions, validating its ability to effectively balance exploration and exploitation. While the proposed HCS-BA algorithm has demonstrated significant improvements in VMP, there are several avenues for future research to further enhance its effectiveness, such as scalability for large-scale cloud data centers, integration with machine learning techniques, and even turning it into a multi-objective optimization problem.

References

[1] P.D. Bharathi, P. Prakash, and V.K.K. Muppavarapu, "Virtual Machine Placement Strategies in Cloud Computing", *2017 Innovations in Power and Advanced Computing Technologies (i-PACT)*, Vellore, India, 2017 (https://doi.org/10.1109/IPACT.2017.8244949).

[2] X.-F. Liu *et al.*, "An Energy Efficient Ant Colony System for Virtual Machine Placement in Cloud Computing", *IEEE Transactions on Evolutionary Computation*, vol. 22, pp. 113–128, 2018 (https://doi.org/10.1109/TEVC.2016.2623803).

[3] M. Masdari, S.S. Nabavi, and V. Ahmadi, "An Overview of Virtual Machine Placement Schemes in Cloud Computing", *Journal of Network and Computer Applications*, vol. 66, pp. 106–127, 2016 (https://doi.org/10.1016/j.jnca.2016.01.011).

[4] S. Mejahed and M. Elshrkawey, "A Multi-objective Algorithm for Virtual Machine Placement in Cloud Environments Using a Hybrid of Particle Swarm Optimization and Flower Pollination Optimization", *PeerJ Computer Science*, vol. 8, art. no. 834, 2022 (https://doi.org/10.7717/peerj-cs.834).

[5] A. Alashaikh, E. Alanazi, and A. AlFuqaha, "A Survey on the Use of Preferences for Virtual Machine Placement in Cloud Data Centers", *ACM Computing Surveys*, vol. 54, pp. 1–39, 2020 (https://doi.org/10.1145/3450517).

[6] F. Lopez-Pires and B. Baran, "Virtual Machine Placement Literature Review", *arXiv*, 2015 (https://doi.org/10.48550/arXiv.1506.01509).

[7] M. Abdel-Basset, L. Abdle-Fatah, and A.K. Sangaiah, "An Improved Levy Based Whale Optimization Algorithm for Bandwidth-efficient Virtual Machine Placement in Cloud Computing Environment", *Cluster Computing*, vol. 22, pp. 8319–8334, 2019 (https://doi.org/10.1007/s10586-018-1769-z).

[8] H.O. Salami, A. Bala, S.M. Sait, and I. Ismail, "An Energy-efficient Cuckoo Search Algorithm for Virtual Machine Placement in Cloud Computing Data Centers", *The Journal of Supercomputing*, vol. 77, pp. 13330–13357, 2021 (https://doi.org/10.1007/s11227-021-03807-3).

[9] A. Gopu and N.N. Venkataraman, "Virtual Machine Placement Using Multi-objective Bat Algorithm with Decomposition in Distributed Cloud: MOBA/D for VMP", *International Journal of Applied Metaheuristic Computing*, vol. 12, pp. 62–77, 2021 (https://doi.org/10.4018/IJAMC.2021100104).

[10] M.A. Al-Abaji, "A Literature Review of Cuckoo Search Algorithm", *Journal of Education and Practice*, vol. 11, 2020 (https://doi.org/10.7176/JEP/11-8-01).

[11] X.-S. Yang, "Bat Algorithm: Literature Review and Applications", *International Journal of Bio-Inspired Computation*, vol. 5, pp. 141–149, 2013 (https://doi.org/10.1504/IJBIC.2013.055093).

[12] A.S. Abohamama and E. Hamouda. "A Hybrid Energy Aware Virtual Machine Placement Algorithm for Cloud Environments", *Expert Systems with Applications*, vol. 150, art. no. 113306, 2020 (https://doi.org/10.1016/j.eswa.2020.113306).

[13] F. Alharbi *et al.*, "An Ant Colony System for Energy-efficient Dynamic Virtual Machine Placement in Data Centers", *Expert Systems with Applications*, vol. 120, pp. 228–238, 2019 (https://doi.org/10.1016/j.eswa.2018.11.029).

[14] S. Walton, O. Hassan, K. Morgan, and M.R. Brown, "Modified Cuckoo Search: A New Gradient Free Optimisation Algorithm", *Chaos, Solitons & Fractals*, vol. 44, pp. 710–718, 2011 (https://doi.org/10.1016/j.chaos.2011.06.004).

[15] A.K. Singh, S.R. Swain, D. Saxena, and C.-N. Lee, "A Bio-inspired Virtual Machine Placement Toward Sustainable Cloud Resource Management", *IEEE Systems Journal*, vol. 17, pp. 3894–3905, 2023 (https://doi.org/10.1109/JSYST.2023.3248118).

[16] D.-M. Zhao, J.-T. Zhou, and K. Li, "An Energy-aware Algorithm for Virtual Machine Placement in Cloud Computing", *IEEE Access*, vol. 7, pp. 55659–55668, 2019 (https://doi.org/10.1109/ACCESS.2019.2913175).

[17] E. Barlaskar, Y.J. Singh, and B. Issac, "Enhanced Cuckoo Search Algorithm for Virtual Machine Placement in Cloud Data Centers", *International Journal of Grid and Utility Computing*, vol. 9, pp. 1–17, (https://doi.org/10.1504/IJGUC.2018.090221).

[18] P. Krishnmoorthy, "Performance Analysis of Hybrid Bat Algorithm and Cuckoo Search Algorithm HB-CSA for Task Scheduling in Mobile Cloud Computing", *SSRN Electronic Journal*, 2021 (https://doi.org/10.2139/ssrn.3997784).

[19] X.-S. Yang and S. Deb, "Cuckoo Search via Levy Flights", *2009 World Congress on Nature & Biologically Inspired Computing (NaBIC)*, Coimbatore, India, 2009 (https://doi.org/10.1109/NABIC.2009.5393690).

[20] A. Gopu *et al.*, "Energy-efficient Virtual Machine Placement in Distributed Cloud Using NSGA-III Algorithm", *Journal of Cloud Computing*, vol. 12, art. no. 124, 2023 (https://doi.org/10.1186/s13677-023-00501-y).

[21] X. Ye, Y. Yin, and L. Lan, "Energy-efficient Many-objective Virtual Machine Placement Optimization in a Cloud Computing Environment", *IEEE Access*, vol. 5, pp. 16006–16020, 2017 (https://doi.org/10.1109/ACCESS.2017.2733723).

Sifeddine Benflis, Ph.D. Student
Department of Computer Science
https://orcid.org/0009-0004-9075-7820
E-mail: sif.benflis@univ-batna2.dz
University of Batna 2, Batna, Algeria
https://univ-batna2.dz

Sonia-Sabrina Bendib, Assoc. Prof.
Department of Computer Science
https://orcid.org/0000-0003-4674-6185
E-mail: ss.bendib@univ-batna2.dz
University of Batna 2, Batna, Algeria
https://univ-batna2.dz

Sedrati Maamar, Assoc. Prof.
Department of Computer Science
https://orcid.org/0000-0003-0444-0399
E-mail: m.sedrati@univ-batna2.dz
University of Batna 2, Batna, Algeria
https://univ-batna2.dz

Fatima Z. Cherhabil, Ph.D. Student
Department of Computer Science
https://orcid.org/0000-0002-6937-5092
E-mail: f.cherhabil@univ-batna2.dz
University of Batna 2, Batna, Algeria
https://univ-batna2.dz

Hanane Merouani, Ph.D. Student
Department of Computer Science
https://orcid.org/0009-0008-0669-8442
E-mail: hanene.merouani@univ-batna2.dz
University of Batna 2, Batna, Algeria
https://univ-batna2.dz

A Hybrid Algorithm for the Synthesis of Distributed Antenna Arrays with Excitation Range Control

Magdy A. Abdelhay

Al-Farqadein University, Basrah, Iraq

https://doi.org/10.26636/jtit.2025.4.2293

Abstract — Excitation coefficients with a low dynamic range ratio (DRR) are advantageous in controlling mutual coupling between the elements of an antenna array. Their use also reduces the output power loss and simplifies the design of the feeding network. In this paper, a hybrid algorithm based on invasive weed optimization and convex optimization for the synthesis of distributed arrays with two subarrays is proposed. Arrays of this type are used in numerous applications, e.g. in aircraft. A constraint is added to the optimization problem to control the DRR of the array's excitation vector. Numerical results are presented for position-only, as well as for position and excitation control approaches. The trade-off between the peak sidelobe ratio and the obtained DRR is illustrated by numerical examples.

Keywords — convex optimization, distributed antenna arrays, dynamic range ratio, invasive weed optimization

1. Introduction

A distributed phased array (DPA) is composed of multiple small-scale arrays, which increases the array's arrangement flexibility and expands the its aperture. DPA with a large aperture offers highly favorable characteristics, such as high directivity and narrow mainlobe width. Due to these features, DPA finds use in many applications in communication systems relying on special layout platforms [1], [2], and in other applications which cater to the high demand for good directivity and great precision with increased degrees of freedom [3], [4].

A DPA is ordinarily a sparse array with nodes that can be placed on independent platforms tens of wavelengths apart. This leads to the appearance of grating lobes in the array's pattern. It is essential in many applications to suppress these grating lobes to avoid problems such as interference from undesired locations.

Many synthesis techniques have been proposed to suppress grating lobes in DPAs [5]–[7]. The synthesis process depends on numerous parameters, including position, excitation, and the number of array elements. Many array pattern synthesis techniques employ global optimization techniques, such as genetic [8], invasive weed optimization (IWO) [9], differential evolution [10], and particle swarm optimization [11] algorithms. Convex optimization has also been widely used to synthesize antenna arrays [12]. Compressive sensing-based approaches have been utilized in [13]–[15] for this purpose as well.

Dynamic range ratio (DRR) is defined as the ratio of the array elements' amplitudes at maximum and minimum values. The DRR of the excitation coefficients is usually high in the synthesized arrays with a low sidelobe level (SLL) [16], [17]. High DRR is undesirable, since it complicates the feeding network and increases its cost. Furthermore, low DRR results in better control of the mutual coupling between antenna elements. Many analytical methods based on popular windows and polynomials, for example Gaussian [18] and ultraspherical windows [19], are used to synthesize array patterns with low DRR. Optimization-based methods, which include the need for low DRR as a design objective, are also used to synthesize arrays with low DRR of the excitations [20], [21].

In [22], a hybrid algorithm for synthesizing a distributed array consisting of two subarrays using differential evolution and convex optimization was proposed. In this proposed method, the differential evolution algorithm is used to find the element positions and the iterative reweighted ℓ_1-norm minimization algorithm is employed to find the optimum weights for a given set of element positions. Unfortunately, the use of iterative reweighted ℓ_1-norm minimization is not necessary, as it is usually relied upon to enhance the sparsity in solutions for optimization problems which use ℓ_1-norm instead of ℓ_0 quasi-norm to minimize the number of non-zero elements in the excitation vector [23]. It is not used to further lower the peak sidelobe level (PSLL), as mentioned in [22].

In the case of the work described in [22], the optimization problem for a given position vector which is obtained using the differential evolution algorithm is convex, and there is no need for any relaxation. Furthermore, the results reported in the paper, i.e. those shown in Tab. 1 in [22], did not satisfy the constraint on the distance between the two subarrays, which should be 30λ instead of the reported 18λ. The work also did not consider the DRR of the excitations of the synthesized array.

In this paper, an algorithm based on IWO and convex optimization is proposed to synthesize distributed arrays consisting of two subarrays, with DRR taken into consideration as well. In the proposed algorithm, IWO is used to find the optimum positions of the array's elements under a constraint

on the distance between the two subarrays and a minimum allowed distance between 2 adjacent array elements.

Convex optimization is used to find the optimum excitation vector for a given set of element positions, which minimizes PSLL, with a constraint aimed at minimizing DRR of the excitations. PSLL of the synthesized array is used as the fitness function for the IWO algorithm. To the best of the author's knowledge, this is the first paper focusing on the synthesis of distributed antenna arrays with constraints on the distance between the sub-arrays and the inter-element spacing between the elements in each sub-array, with dynamic range ratio considerations accounted for as well.

The remainder of the paper is organized as follows. Section 2 formulates the problem. The proposed algorithm is detailed in Section 3. Numerical examples are given in Section 4, and conclusions are drawn in Section 5.

2. Synthesis Problem Formulation

Consider a linear array made up of two identical sub-arrays which consist of $2 \times M$ isotropic radiating elements, with the distance between the sub-arrays equaling D_0. The distance between the individual elements in the same subarray is d_0. The location of the n-th array element x_n can be expressed as:

$$x_n = \begin{cases} -(N-n)\,d_0 - \frac{D_0}{2}, & 1 \leqslant m \leqslant M \\ \frac{D_0}{2} + (n-N-1)\,d_0, & N+1 \leqslant n \leqslant 2N \end{cases} . \tag{1}$$

The distance between two elements on the left-hand side of each subarray equals:

$$x_{N+1} - x_1 = D_0 + (N-1)\,d_0\,. \tag{2}$$

The array's far field pattern can be written as:

$$AF(\theta) = \sum_{n=1}^{2N} w_n\, e^{-jkx_n \sin\theta}\,, \tag{3}$$

where w_n is the excitation of the n-th element, $k = \frac{2\pi}{\lambda}$ is the wave number, λ is the wavelength, and θ is the elevation angle. Equation (3) can be written in a matrix form as:

$$AF(\theta) = \boldsymbol{A}(\theta)^T \boldsymbol{w}\,, \tag{4}$$

where T is the transpose operator,

$$\boldsymbol{A}(\theta) = \left[e^{-jkx_1 \sin\theta},\ e^{-jkx_2 \sin\theta}, \ldots,\ e^{-jkx_{2N} \sin\theta}\right]^T$$

and

$$\boldsymbol{w} = [w_1,\ w_2,\ \ldots,\ w_{2N}]^T\ .$$

The objective here is to find element locations and excitations that minimize the peak sidelobe level (PSLL), subject to constraints on the number of elements, minimum element separation, and a fixed distance between the two sub-arrays.

Mathematically, the optimization problem can be expressed as:

$$\begin{cases} \text{find } \boldsymbol{x} = [x_1, \ldots, x_{2N}]^T \text{ and } \boldsymbol{w} = [w_1, \ldots, w_{2N}]^T \\ \min\{\text{PSLL}(\boldsymbol{x}, \boldsymbol{w}), \text{DRR}(\boldsymbol{w})\} \\ \text{subject to } \ x_{i+1} - x_i \geqslant d_c > 0 \\ \qquad\qquad i \in \mathbb{Z},\ 1 \leqslant i \leqslant 2N-1,\ i \neq N \\ \qquad\qquad x_{N+1} - x_N \geqslant D_0 > 0 \\ \qquad\qquad x_0 = 0 \end{cases}, \tag{5}$$

where d_c is the minimum allowable distance between elements in each sub-array and the PSLL is defined as:

$$\text{PSLL}(\boldsymbol{x}, \boldsymbol{w}) = \max \left| \frac{\sum_{n=1}^{2N} w_n\, e^{-jkx_n \sin(\theta_{\text{sl}})}}{AF(\theta_0)} \right|, \tag{6}$$

where θ_0 is the direction of the mainlobe, θ_{sl} is the sidelobe angles outside of the mainlobe region, and $|\cdot|$ is the absolute value.

The DRR is defined as:

$$\text{DRR} = \frac{\max\{|\,w_k\,|\}}{\min\{|\,w_k\,|\}}, \qquad k = 1, 2, \ldots, 2N\,, \tag{7}$$

which represents the ratio of maximum and minimum values for the amplitudes of the array's elements.

3. The Proposed Hybrid Method

Hybrid IWO and convex optimization algorithms are used to solve the optimization problem in Eq. (5). The proposed algorithm is summarized below.

3.1. Element Position Initialization

The individual here is taken as the position vector $\boldsymbol{x} = [x_1, \ldots, x_{2N}]^T$. For the sake of satisfying the constraints on the minimum spacing between the elements in each subarray d_0 and the space between the two subarrays D_0, the position vector is expressed as follows:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_N \\ x_{N+1} \\ x_{N+2} \\ \vdots \\ x_{2N} \end{bmatrix} = \begin{bmatrix} 0 \\ a_1 \\ a_2 \\ \vdots \\ a_{N-1} \\ a_N \\ a_{N+1} \\ \vdots \\ a_{2N-1} \end{bmatrix} + \begin{bmatrix} 0 \\ d_0 \\ 2\,d_0 \\ \vdots \\ (N-1)\,d_0 \\ (N-1)\,d_0 + D_0 \\ N d_0 + D_0 \\ \vdots \\ (2N-2)\,d_0 + D_0 \end{bmatrix} \tag{8}$$

The vector $\boldsymbol{a} = [a_1, \ldots, a_{2N-1}]^T$ consists of $2N-1$ real random numbers in the range of $[0, \text{V}_{\text{max}}]$, and elements of $\boldsymbol{a}$ are ordered in ascending order, i.e. $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_{2N-1}$.

The position vector $\boldsymbol{x}$ can be determined after generating $\boldsymbol{a}$. Here $\boldsymbol{a}$ is considered the seed for the IWO algorithm. By producing $\boldsymbol{a}$ M times independently, a starting population of M seeds is initialized. Consequently, a set of M position vectors are initialized.

3.2. Fitness Function

Provided that the positions of the array elements are determined by the IWO algorithm, the optimization problem in Eq. (5) is a convex optimization problem which can be solved efficiently using off-the-shelf packages, such as CVX [24]. In such a case, the optimization problem can be expressed mathematically as:

$$\min_{w,\tau_s} \tau_s \tag{9a}$$

$$\text{subject to } \operatorname{Re}\left\{\boldsymbol{A}(\theta_0)^T\boldsymbol{w}\right\} = \tau_m \tag{9b}$$

$$\left|\boldsymbol{A}(\theta_{\text{sl}})^T\boldsymbol{w}\right| \leqslant \tau_s \tag{9c}$$

$$\|\boldsymbol{w}\| \leqslant \tau_d \tag{9d}$$

where $\operatorname{Re}\{\cdot\}$ is the real part.

Without normalization, τ_m is the directivity of original distributed array and τ_s is a slack variable which represents an upper bound on the response of the array in the sidelobe region. $\|\cdot\|$ is the ℓ_2-norm, which is the square root of the sum of the squared values of the vector elements. τ_d represents an upper on the ℓ_2-norm of the excitation vector $\boldsymbol{w}$.

Unlike the ℓ_1-norm, the ℓ_2-norm does not promote sparsity in solutions. Instead, it distributes the penalty across all coefficients, resulting in more evenly distributed values. This leads to a reduction in the ratio between the largest and smallest values that element excitations can assume, which results in a decrease in the DRR.

The resulting PSLL of the array is considered to be the fitness value of the correspondent seed in the population. Every initial seed grows into a weed after calculating its fitness.

3.3. Reproduction

The reproductive capability of weeds depends on their fitness values. A linear relationship exists between the number of seeds reproduced from every weed and its fitness value, i.e. PSLL associated with the weed. Here, the weeds with lower fitness values have a larger probability of being preserved in the population and, hence, produce more seeds. The number of seeds produced by the m-th weed can be expressed as:

$$s_m = \frac{S_{\text{max}} - S_{\text{min}}}{f_{\text{max}} - f_{\text{min}}}(f_{\text{max}} - f_{\text{m}}) + S_{\text{min}} \,, \tag{10}$$

where f_{max} and f_{min} are the maximum and minimum fitness values, i.e. PSLLs, in the current population, respectively. S_{max} and S_{min} are the maximum and minimum allowable seeds, respectively. f_{m} is the fitness value of the m-th weed.

3.4. Spatial Dispersal

New seeds are then dispreaded in a random manner over the searching space. Gaussian distribution is used with mean μ equal to the location of the parent weed. During the iterations,

Tab. 1. List of element positions for $N = 25$ element array with position-only control.

n	Pos. (λ)	n	Pos. (λ)	n	Pos. (λ)	n	Pos. (λ)
1	0	14	12.3381	27	40.7146	40	49.7632
2	0.7003	15	12.9280	28	41.2152	41	50.8395
3	1.3394	16	14.3832	29	41.9873	42	52.4788
4	1.8571	17	15.4114	30	42.7831	43	53.1647
5	3.0920	18	16.2914	31	43.3900	44	54.0020
6	3.6944	19	16.8024	32	43.9927	45	54.7090
7	4.7633	20	17.5651	33	44.5261	46	56.7148
8	5.5138	21	18.1179	34	45.0555	47	57.3373
9	6.7853	22	18.6586	35	45.5993	48	58.0154
10	7.5022	23	19.1612	36	46.1893	49	58.7423
11	8.2622	24	19.7004	37	47.1063	50	59.3086
12	9.0797	25	20.2101	38	48.4132		
13	11.4735	26	40.2112	39	49.1710		

the standard deviation σ is reduced from its initial maximum value σ_{initial} to its final minimum value σ_{final}. The value of σ during iteration i can be calculated using the relation:

$$\sigma = \frac{(i_{\text{max}} - i)^n}{(i_{\text{m}})^n}(\sigma_{\text{initial}} - \sigma_{\text{final}}) - \sigma_{\text{final}} \,, \tag{11}$$

where n is a nonlinear modulation index and i_{max} is the maximum number of iterations.

Then the k-th seed produced by the m-th weed may be written as:

$$\boldsymbol{a}_{n,k} = \boldsymbol{a}_n + \mathcal{N}(0, \sigma) \,. \tag{12}$$

Following that, the elements of each seed $\boldsymbol{a}$ are limited in the range of $[0, \text{V}_{\text{max}}]$ and thus ordered in an increasing order $a_1 \leqslant a_2 \leqslant \ldots \leqslant a_{2N-1}$.

Equation (8) is thus used to calculate the corresponding position vector, and next (9) is used to find the optimum excitation vector which minimizes the PSLL of the distributed array pattern.

3.5. Competitive Exclusion

The weeds are grown from the seeds and ranked together with parent weeds based on their PSLL fitness value. As the number of weeds increases, there must be some sort of competition between them to limit their maximum number in the colony. When the maximum number of weeds p_{max} is reached, weeds with poor fitness, i.e., with their PSLL being high in comparison to that of other weeds, are removed from the current colony. On the other hand, the weeds with better fitness will survive and be allowed to reproduce their next generations. The process is repeated as described in Subsection 3.3 until the termination process criteria are met, i.e., the number of maximum iterations i_{max} is reached.

4. Simulation Results

4.1. Position-only Control

Consider an array of 50 elements, which consists of two subarrays, each containing $N = 25$ elements. The distance

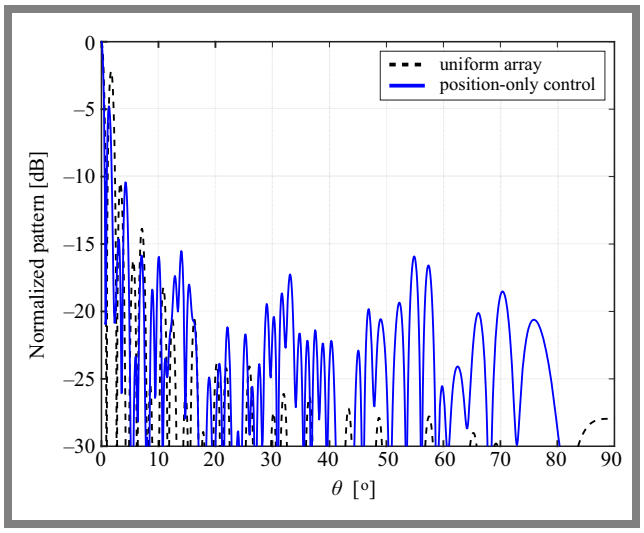

Fig. 1. Patterns of the original uniformly spaced array vs. the array with uniform amplitudes and optimized element positions.

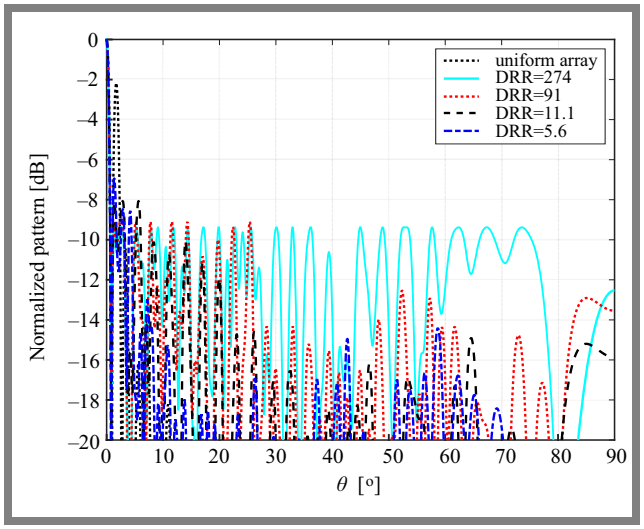

Fig. 2. Patterns of the synthesized arrays with with $D_0 = 20\lambda$.

between subarrays is $D_0 = 20\lambda$, and the distance between the elements in each subarray is $d_0 = 0.5\lambda$. The array with uniform amplitudes and fixed spacing between the elements has a PSLL of –2.17 dB for the normalized pattern. Optimizing only the positions of the array elements resulted in an array with a PSLL of –4.79 dB for the normalized pattern. The first null beam width (FNBW) of array pattern is 1.8°. A list of the position of each element is given in Tab. 1.

It can be seen from the list that the distance between each successive elements is greater than or equal to $d_0 = 0.5\lambda$ and the distance between the two subarrays equals to $D_0 = 20\lambda$. Therefore, the constraints on the optimization problem are satisfied in the synthesized array. The normalized patterns of the uniformly spaced array and the synthesized array with optimized element locations are depicted in Fig. 1.

4.2. Position and Excitation Control

The same array as described in Subsection 4.1 ($N = 25$, $d_0 = 0.5\lambda$ and $D_0 = 20\lambda$) is considered here. The array is

Tab. 2. List of element positions and normalized excitations for $N = 25$ element array with no constraint of the weight vector $\boldsymbol{w}$.

n	Position (λ)	w_n	n	Position (λ)	w_n
1	0	0.0454	26	37.1519	1.0000
2	0.5338	0.0867	27	38.3303	0.1541
3	1.3945	0.0710	28	39.3940	0.2147
4	2.0561	0.0518	29	40.2403	0.0037
5	2.6577	0.0095	30	41.0885	0.0000
6	3.3152	0.0000	31	42.5882	0.1567
7	3.9472	0.0941	32	43.7477	0.1770
8	4.4813	0.0647	33	44.2883	0.0836
9	5.2003	0.0645	34	45.9830	0.0000
10	5.8500	0.0591	35	46.5271	0.0782
11	6.8806	0.0000	36	47.0757	0.1670
12	7.7895	0.1113	37	47.6219	0.0419
13	8.6952	0.1279	38	48.9694	0.0489
14	9.4475	0.1163	39	49.7959	0.0700
15	9.9618	0.0000	40	50.9733	0.1224
16	10.9504	0.0000	41	52.3476	0.0000
17	11.5424	0.0887	42	52.9467	0.0604
18	12.3939	0.1194	43	53.7691	0.0000
19	12.9026	0.0000	44	54.8033	0.0952
20	13.5500	0.0000	45	55.5401	0.0836
21	14.0602	0.0000	46	56.1475	0.1136
22	14.9208	0.0312	47	56.7744	0.0000
23	15.6179	0.0000	48	57.3970	0.0000
24	16.5354	0.3159	49	58.2532	0.0774
25	17.1519	0.4404	50	60.9913	0.4737

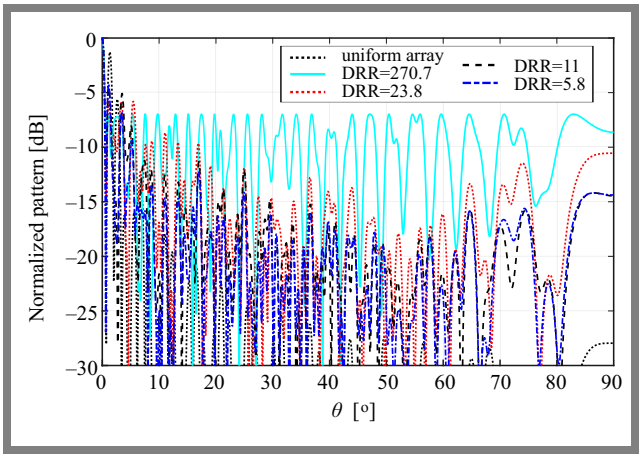

Fig. 3. Patterns of the synthesized arrays with $D_0 = 30\lambda$.

synthesized by optimizing both the positions and excitations of the array elements. We start with optimizing the array using the objective function given in (9a) under the constraints defined Eqs. (9b) and (9c) only. That is, there is no constraint on the the ℓ_2-norm of the weight vector $\boldsymbol{w}$. The resultant array has a PSLL of –9.38 dB with a DRR of 274. Table 2 contains

Tab. 3. List of element positions and normalized excitations for $N = 25$ element array with $\tau_d = 10$.

n	Position (λ)	w_n	n	Position (λ)	w_n
1	0	0.2012	26	36.4994	1.0000
2	0.8820	0.1335	27	36.9994	0.7551
3	1.4606	0.1050	28	37.5001	0.5570
4	2.0104	0.0899	29	39.2464	0.2014
5	2.5173	0.0856	30	39.8575	0.1748
6	3.4254	0.0908	31	40.3841	0.1801
7	3.9456	0.0970	32	41.4258	0.2285
8	4.4538	0.1055	33	42.1098	0.2608
9	5.1479	0.1204	34	44.2116	0.2098
10	5.8642	0.1315	35	45.0513	0.1605
11	6.9215	0.1346	36	45.6693	0.1337
12	7.4267	0.1365	37	46.2975	0.1179
13	8.3457	0.1447	38	46.8566	0.1134
14	9.3621	0.1563	39	47.4929	0.1144
15	10.3648	0.1478	40	48.0668	0.1159
16	10.9275	0.1238	41	48.8481	0.1191
17	11.4564	0.0926	42	50.5966	0.1194
18	12.0924	0.0540	43	51.3460	0.1136
19	12.8909	0.0174	44	52.6031	0.1268
20	13.5487	0.0110	45	53.5261	0.1574
21	14.4040	0.0640	46	54.2029	0.1809
22	14.9987	0.1582	47	55.5386	0.1786
23	15.4989	0.2835	48	56.0452	0.1657
24	15.9991	0.4580	49	57.9896	0.2040
25	16.4993	0.6843	50	60.5225	0.9048

a list of element positions and the corresponding normalized weights.

Next the optimization problem in (9) is considered under all the constraints. The value of τ_d is set to 10 experimentaly. After optimizing the pattern using the proposed hybrid IWO and convex optimization algorithm, the optimized pattern has a PSLL of –9.10 dB and DRR = 91. The PSLL increased by 0.28 dB (3%) and the DRR decreased by 183 (66.79%) compared to the unconstrained $\|\boldsymbol{w}\|$. The trade-off is obvious between the PSLL and the DRR and will be more obvious as we decrease the value of τ_d. A list of the element position the their normalized excitation is given in Tab. 3.

Next the algorithm is run with $\tau_d = 9$. The obtained PSLL of the normalized pattern is –8.07 dB with DRR of 11.1. This corresponds to an increase in the PSLL of 1.31 (14%) and a decrease in the DRR by –262.9 (96%) compared to the case of unconstrained $\|\boldsymbol{w}\|$. Table 4 lists element positions and the corresponding normalized weights.

Tab. 4. List of element positions and normalized excitations for $N = 25$ element array with $\tau_d = 9$.

n	Position (λ)	w_n	n	Position (λ)	w_n
1	0	0.7657	26	41.0850	0.7198
2	0.5136	0.6078	27	41.5991	0.5804
3	1.1969	0.4334	28	42.3679	0.4144
4	2.5289	0.2078	29	43.1618	0.2939
5	3.7192	0.1142	30	44.2219	0.2020
6	4.2407	0.0971	31	44.7751	0.1786
7	4.9250	0.0902	32	45.5175	0.1657
8	7.0583	0.1243	33	46.6976	0.1701
9	7.8754	0.1425	34	47.5546	0.1794
10	8.5126	0.1553	35	48.1616	0.1848
11	9.3579	0.1694	36	48.7338	0.1875
12	9.9082	0.1769	37	49.6791	0.1863
13	11.3564	0.1896	38	50.3039	0.1817
14	12.3723	0.1920	39	50.9258	0.1748
15	13.3857	0.1890	40	51.5208	0.1666
16	14.7962	0.1812	41	52.3721	0.1530
17	15.3058	0.1807	42	53.0022	0.1423
18	16.1448	0.1893	43	53.6061	0.1319
19	16.7389	0.2072	44	54.4423	0.1184
20	17.5210	0.2531	45	55.0639	0.1104
21	18.3334	0.3374	46	55.6404	0.1060
22	19.3443	0.5083	47	56.5578	0.1101
23	20.0494	0.6770	48	57.3778	0.1328
24	20.5683	0.8282	49	58.2990	0.1918
25	21.0819	1.0000	50	58.9904	0.2677

Finally the algorithm is run for $\tau_d = 8$. The obtained normalized pattern has a PSLL of –6.9 dB and DRR of 5.6. This corresponds to an increase in the PSLL by 2.48 (26.4%) and a decrease in DRR by 268.41 (98%) compared to the case of unconstrained $\|\boldsymbol{w}\|$. A list of the element positions and their normalized excitations are given in Tab. 5. Again, the trade-off is clear between the obtained PSLL and the resultant DRR. It is also obvious that as the $\|\boldsymbol{w}\|$ is constrained to has a lower value, the value of the resultant DRR improves (decreased). The patterns of the three cases of τ_d (i.e. $\|\boldsymbol{w}\|$) are shown in Fig. 2.

4.3. Effect of Distance Between Subarrays

In this section, the distance between the two subarrays is increased to 30λ. It is expected that as the distance between the subarrays increases, the grating lobe level will increase and the FNBW will decrease. For the uniform array with $D_0 = 30\lambda$, the PSLL is –1.27 dB compared to –2.17 dB for

Tab. 5. List of element positions and normalized excitations for $N = 25$ element array with $\tau_d = 8$.

n	Position (λ)	w_n	n	Position (λ)	w_n
1	0	0.4989	26	37.9738	1.0000
2	0.5024	0.4563	27	38.5722	0.9375
3	1.1273	0.4066	28	39.2448	0.8669
4	1.6606	0.3674	29	39.8553	0.8031
5	2.3929	0.3187	30	40.5724	0.7291
6	3.0146	0.2825	31	41.2286	0.6630
7	3.7743	0.2451	32	42.5078	0.5408
8	4.3987	0.2203	33	43.1215	0.4864
9	6.0796	0.1819	34	43.8583	0.4255
10	6.8959	0.1786	35	44.5376	0.3742
11	7.4659	0.1823	36	45.2974	0.3229
12	7.9836	0.1900	37	46.1382	0.2744
13	8.5192	0.2022	38	46.9678	0.2358
14	9.0566	0.2187	39	48.6926	0.1870
15	9.7858	0.2478	40	49.3271	0.1803
16	10.3167	0.2736	41	50.0797	0.1802
17	11.0325	0.3143	42	51.9051	0.2158
18	11.7437	0.3610	43	52.9070	0.2559
19	12.8468	0.4448	44	54.1803	0.3264
20	13.3873	0.4903	45	55.0326	0.3847
21	14.4320	0.5851	46	55.8927	0.4514
22	15.6409	0.7038	47	56.5112	0.5037
23	16.2322	0.7643	48	57.6209	0.6052
24	16.8421	0.8277	49	58.4899	0.6899
25	17.8622	0.9347	50	59.1051	0.7519

the array with $D_0 = 20\lambda$, and the FNBW is 1.4° compared to 1.8° for the array with $D_0 = 20\lambda$. The array is optimized using the proposed algorithm by optimizing both the positions and weights of the array elements for different values of τ_d.

For unconstrained $\|\boldsymbol{w}\|$, the obtained PSLL is –6.9770 and the DRR is 270.73. For the case with $\tau_d = 10$, the PSLL is –5.8153 and DRR is 23.79. For $\tau_d = 9$, the PSLL is –5.12 and DRR equals 11. Finally, for $\tau_d = 8$, the PSLL equals –4.4039 and the DRR is 5.8. Figure 3 shows the pattern of the uniform array alongside the patterns for the different obtained DRRs. Table 6 summarizes the obtained results. From Tab. 6, it can be seen that as the distance between the sub-arrays increases, the performance of the array deteriorates.

5. Conclusion

An algorithm based on IWO and convex optimization was presented. The algorithm optimizes the elements' positions and excitations in distributed arrays with two subarrays. Numerical results showed a clear trade-off between the obtained PSLL and the value of DRR. Low DRR resulted in higher PSLL and vice versa.

Tab. 6. PSLL of optimized arrays with different distances between subarrays.

Distance		$D_0 = 20\,\lambda$	$D_0 = 30\,\lambda$
Uniform		–2.17 dB	–1.27 dB
$\tau = \infty$	PSLL	–9.38 dB	–6.977 dB
	DRR	274	270.7
$\tau_d = 10$	PSLL	–9.1 dB	–5.815 dB
	DRR	91	23.8
$\tau_d = 9$	PSLL	–8.07 dB	–5.123 dB
	DRR	11.1	11
$\tau_d = 8$	PSLL	–6.9 dB	–4.404 dB
	DRR	5.6	5.8

References

[1] S. Rao, A. Pandya, and C. Ostroot, "Phased Array Antennas for Aircraft Applications", *2018 IEEE Indian Conference on Antennas and Propagation (InCAP)*, Hyderabad, India, 2018 (https://doi.org/10.1109/INCAP.2018.8770894).

[2] C. Loecker, P. Knott, R. Sekora, and S. Algermissen, "Antenna Design for a Conformal Antenna Array Demonstrator", *2012 6th European Conference on Antennas and Propagation (EUCAP)*, Prague, Czech Republic, 2012 (https://doi.org/10.1109/EuCAP.2012.6206004).

[3] M. Devipriya and M. Brindha, "Moving Object Tracking Using FPGA", *2017 International Conference on Intelligent Sustainable Systems (ICISS)*, Palladam, India, 2017 (https://doi.org/10.1109/ISS1.2017.8389454).

[4] P. Nayeri, "Focused Antenna Arrays for Wireless Power Transfer Applications", *2018 International Applied Computational Electromagnetics Society Symposium (ACES)*, Denver, USA, 2018 (https://doi.org/10.23919/ROPACES.2018.8364266).

[5] B.-K. Feng and D. C. Jenn, "Two-way Pattern Grating Lobe Control for Distributed Digital Subarray Antennas", *IEEE Transactions on Antennas and Propagation*, vol. 63, pp. 4375–4383, 2015 (https://doi.org/10.1109/TAP.2015.2465863).

[6] R. Liu *et al.*, "Transmit-receive Beamforming for Distributed Phased-MIMO Radar System", *IEEE Transactions on Vehicular Technology*, vol. 71, pp. 1439–1453, 2022 (https://doi.org/10.1109/TVT.2021.3133596).

[7] Y. Wang, Q. Yang, H. Wang, and Y. Zeng, "Grating Lobe Suppression for Distributed Phased Array via Accumulated Array Pattern Synthesis", *IEEE Antennas and Wireless Propagation Letters*, vol. 22, pp. 1527–1531, 2023 (https://doi.org/10.1109/LAWP.2023.3249908).

[8] J.R. Mohammed and D.A. Aljaf, "Joint Optimization of Sum and Difference Patterns with a Common Weight Vector Using the Genetic Algorithm", *Journal of Telecommunications and Information Technology*, no. 3, pp. 67–73, 2022 (https://doi.org/10.26636/jtit.2022.160722).

[9] S. Pal, A. Basak, S. Das, and A. Abraham, "Linear Antenna Array Synthesis with Invasive Weed Optimization Algorithm", *2009 International Conference of Soft Computing and Pattern Recognition*, Malacca, Malaysia, 2009 (https://doi.org/10.1109/SoCPaR.2009.42).

[10] H. Singh *et al.*, "End Fire Linear Antenna Array Synthesis Using Differential Evolution Inspired Adaptive Naked Mole Rat Algorithm",

Scientific Reports, vol. 13, art. no. 12308, 2023 (https://doi.org/10.1038/s41598-023-39509-4).

[11] E.R. Schlosser, S.M. Tolfo, and M.V.T. Heckler, "Particle Swarm Optimization for Antenna Arrays Synthesis", *2015 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)*, Porto de Galinhas, Brazil, 2015 (https://doi.org/10.1109/IMOC.2015.7369120).

[12] M.A. Abdelhay and S.E. El-Khamy, "A Hybrid Algorithm for the Synthesis of Sparse Concentric Ring Arrays", *2024 41st National Radio Science Conference (NRSC)*, New Damietta, Egypt, 2024 (https://doi.org/10.1109/NRSC61581.2024.10510470).

[13] J.R. Mohammed, R.H. Thaher, and A.J. Abdulqader, "Linear and Planar Array Pattern Nulling via Compressed Sensing", *Journal of Telecommunications and Information Technology*, no. 3, pp. 50–55, 2021 (https://doi.org/10.26636/jtit.2021.152921).

[14] M.A. Abdelhay and S.E. El-Khamy, "A Compressed Sensing-based Approach for Null Steering in Partially Adaptive Planar Arrays Using a Reduced Number of Adjustable Array Elements", *Digital Signal Processing*, vol. 145, art. no. 104311, 2024 (https://doi.org/10.1016/j.dsp.2023.104311).

[15] S.E. El-Khamy, N.O. Korany, and M.A. Abdelhay, "Minimising Number of Perturbed Elements in Linear and Planar Adaptive Arrays with Broad Nulls Using Compressed Sensing Approach", *IET Microwaves, Antennas & Propagation*, vol. 13, pp. 1134–1141, 2019 (https://doi.org/10.1049/iet-map.2018.5221).

[16] R. Vescovo, "Consistency of Constraints on Nulls and on Dynamic Range Ratio in Pattern Synthesis for Antenna Arrays", *IEEE Transactions on Antennas and Propagation*, vol. 55, pp. 2662–2670, 2007 (https://doi.org/10.1109/TAP.2007.905828).

[17] G.K. Mahanti, A. Chakraborty, and S. Das, "Design of Fully Digital Controlled Reconfigurable Array Antennas with Fixed Dynamic Range Ratio", *Journal of Electromagnetic Waves and Applications*, vol. 21, pp. 97–106, 2007 (https://doi.org/10.1163/156939307779391768).

[18] G. Buttazzoni and R. Vescovo, "Gaussian Approach versus Dolph-Chebyshev Synthesis of Pencil Beams for Linear Antenna Arrays", *Electronics Letters*, vol. 54, pp. 8–10, 2018 (https://doi.org/10.1049/el.2017.3098).

[19] S.W.A. Bergen and A. Antoniou, "Design of Ultraspherical Window Functions with Prescribed Spectral Characteristics", *EURASIP Journal on Advances in Signal Processing*, vol. 2004, art. no. 196503, 2004 (https://doi.org/10.1155/S1110865704403114).

[20] F.E.S. Santos and J.A.R. Azevedo, "Adapted Raised Cosine Window Function for Array Factor Control with Dynamic Range Ratio Limitation", *2017 11th European Conference on Antennas and Propagation (EUCAP)*, Paris, France, 2017 (https://doi.org/10.23919/EuCAP.2017.7928824).

[21] G. Molnar and M. Matijascic, "Gegenbauer Arrays with Minimum Dynamic Range Ratio and Maximum Beam Efficiency", *2020 IEEE International Symposium on Antennas and Propagation and North American Radio Science Meeting*, Montreal, Canada, 2020 (https://doi.org/10.1109/IEEECONF35879.2020.9330164).

[22] S. Fang, W. Li, Z. Xue, and W. Ren, "Synthesis of Distributed Array Consisting of Two Subarrays via Hybrid Method of Differential Evolution Optimization and Convex Optimization", *IEEE Antennas and Wireless Propagation Letters*, vol. 20, pp. 125–129, 2021 (https://doi.org/10.1109/LAWP.2020.3035177).

[23] E.J. Candès, M.B. Wakin, and S.P. Boyd, "Enhancing Sparsity by Reweighted $\ell 1$ Minimization", *Journal of Fourier Analysis and Applications*, vol. 14, pp. 877–905, 2008 (https://doi.org/10.1007/s00041-008-9045-x).

[24] M. Grant and S.P. Boyd, "CVX: Matlab Software for Disciplined Convex Programming", 2014.

Magdy A. Abdelhay, Ph.D., Assistant Professor
Department of Information and Communication Engineering
https://orcid.org/0000-0003-4244-0840
E-mail: abdelhay@ieee.org
Al-Farqadein University, Basrah, Iraq
https://www.fu.edu.iq/en

Blockchain-implied Architecture for Secure and Energy Efficient Processing of IoT Data in Pervasive WSNs

Sushovan Das[1] and Uttam Kr. Mondal[2]

[1]*College of Engineering and Management, Kolaghat, Purba Medinipur, India,*
[2]*Vidyasagar University, Midnapur, India*

https://doi.org/10.26636/jtit.2025.4.2194

Abstract — Pervasive wireless sensor networks (PWSNs) are essential for real-time data transmission in Internet of Things (IoT) environments. However, conventional centralized models, while energy efficient, often face challenges related to data integrity and security. This paper proposes a decentralized blockchain-based architecture aimed at enhancing secure IoT data processing at the base station while preserving energy efficiency. The system utilizes a blockchain network among sink nodes and its operation is divided into four stages: deployment of a virtual machine on leaf nodes for real-time data collection, generation of hash keys to ensure secure transmission to sink nodes, implementation of a universal virtual machine (UVM) at the sink layer for block formation, and development of an integrated authentication and consensus module within the UVM. The proposed framework ensures efficient, verifiable and efficient data handling. Performance is evaluated using sensor node energy efficiency (SNEN), blockchain energy consumption level (BCLE), blockchain transmission efficiency (BCTE), and packet delivery in sink nodes (PDSN). Experimental results demonstrate improved energy efficiency in the sensor zone, reduced blockchain latency, and improved throughput, establishing a robust and secure model for data handling in PWSNs.

Keywords — blockchain, data integration, energy efficiency, Internet of Things, pervasive WSN

1. Introduction

The huge growth in the number of Internet of Things (IoT) devices has caused an unprecedented increase in the generation of data collected from such environments as smart cities, healthcare, agriculture and industrial automation. However, the management and integration of these heterogeneous data sets is a difficult task due to issues concerning data integrity, security, and interoperability. To address these issues, the use of blockchain technology [1], [2] combined with a cryptographic hash key [3], [4] is a promising solution.

Blockchain technology provides secure data transmission through a complex encryption system [5], similar to a meticulous accounting ledger of a company. It carefully monitors and records all transactions on a peer-to-peer network. Each block in the chain contains data about its creation time and is connected to the previous block through a unique hash code and transaction details. Once recorded on the network, the data is immutable. Blockchain is designed to prevent fraud and data tampering attempts. It requires complex computational processes, such as data encryption [6], [7] and decryption in a distributed environment, leading to higher energy consumption compared to the conventional approach of a centralized network structure often used in WSNs. However, the centralized network structure may compromise the security of data integration.

This paper proposes a hybrid model for pervasive wireless sensor networks (PWSN) that utilizes a decentralized network structure [8]–[10]. The PWSN is divided into multiple sensor zones, each with a sink node. These sink nodes are connected through a cloud environment, forming a distributed network. Within each sensor zone, a centralized network is replicated, with sensor nodes acting as leaf nodes. These leaf nodes are responsible for data preparation tasks such as hash generation, compression, and transmission to the respective sink node, thus reducing energy consumption compared to a fully distributed blockchain implementation. To address energy and computational complexity issues, the sink nodes, which are high-end computers powered by the mains, participate in the creation and integration of blockchain data for the entire PWSN. The model addresses the challenges of trustworthiness, privacy, and interoperability of IoT data using blockchain technology and hash functions.

The key contributions of this research are as follows:

- Design of a decentralized architecture for PWSN.
- Developing algorithms for energy-efficient data collection and preparation using semantic technology at the leaf nodes.
- Incorporating cryptographic hash functions before data transmission to sink nodes.
- Designing a universal virtual machine (UVM) for sink nodes to manage data storage and blockchain integration.
- Integrating authentication and consensus modules within the UVM for secure block validation.

2. Literature Survey

The authors of [11] conducted an in-depth survey of blockchain technology, including its history, consensus algo-

rithms, cryptography, and various blockchain applications. The work also emphasized blockchain security, covering risk analysis, security risk categories, real attacks, bug analysis, and recent security measures. Article [12] investigated the integration of blockchain technology into wireless sensor networks (WSN), highlighting its advantages and potential challenges.

The authors of [13] showed the use of the blockchain technology to improve the security of WSNs. This research smoothly incorporated blockchain into data transfer processes, forming a highly secure structure for WSNs. By implementing a blockchain-based transaction ledger, sensor data is converted into unalterable records. The new system they proposed is excellent in the aggregation and analysis of sensor data, significantly increasing the reliability of the entire wireless sensing network architecture.

As a result, the study concluded that blockchain technology can be an effective solution to the security problem of distributed storage data. In [14], a thorough examination of the role of blockchain in the metaverse is conducted. The authors presented the basic principles of blockchain and the metaverse to demonstrate how blockchain solutions can address issues such as storage, integrity, security, and interoperability.

In [15], the authors suggest a way to strengthen data security in WSNs by incorporating blockchain into data transmission, leading to a highly secure wireless sensor network. Paper [16] presented a novel blockchain-based architecture to address the storage issues of massive IoT data. This decentralized system uses blockchain immutability, security, transparency, and automation, providing reliable data management.

The research demonstrated remarkable results, making it a scalable solution for IoT data storage. It is protocol-agnostic, allowing easy integration into various IoT applications, and revolutionizing data management in the IoT domain.

The survey conducted in [17] starts by introducing traditional WSN solutions and then delves into how blockchain technology can be used to improve data management. It also looks at the important role of blockchain in strengthening security. It begins by examining centralized WSN models and the security issues they face. After that, a thorough investigation of blockchain-based WSN solutions is presented, designed to address various security aspects, such as access control, preservation of information integrity, assurance of privacy, and extension of the longevity of WSN nodes.

Researchers in [18] investigated the impact of security protocols on wireless sensor networks and their ability to collect and analyze data. This study offers a concise overview of data aggregation and data compression using blockchain technology in WSNs for secure data transmission. The authors of [19] combined blockchain and IoT to create a distributed storage architecture that would protect the integrity and security of WSN data storage in edge computing. Paper [20] provides an overview of the protocols used to integrate blockchain technology with IoT. The researchers studied the consensus protocols used to create blockchains for IoT applications.

The study in [21] introduces a blockchain-based incentive system for WSNs. This system uses two blockchains: one to store node data and the other to manage data access. To reduce the storage requirements of network nodes, the preserving hash functions are used to compare stored data with new data blocks, and the latter are stored in nodes closest to existing data. The authors of [22] suggested a cryptographic iterative hash function system to improve the security of WSNs when transmitting sensor data on the blockchain. To improve sensor security, the Merkle tree algorithm was used in the investigation.

In [23], the authors proposed a hybrid blockchain-based model that incorporates a mutual authentication scheme to identify the cluster head node in a pervasive wireless sensor environment. In [24], a new and effective authentication system is proposed that uses blockchain technology to improve security in WSNs, essential components of the IoT network. The proposed approach established a hierarchical blockchain network with local and global chains, allowing secure connections among nodes in various communication scenarios, such as user authentication, identity verification, and cluster node validation.

The authors of [25] proposed a decentralized blockchain-based system that integrates authentication and privacy protocols for secure communication in WSNs enabled by the Internet of Things. This system includes a registration, certification, and revocation process for secure communication between sensor nodes and the central base station (BS) in a cloud environment. The performance of the solution was evaluated on metrics such as detection accuracy, certification delay, and computational and communication overhead.

In [26], a blockchain-based system is proposed for registration, authentication, data sharing, and non-repudiation in the Internet of Wireless Sensor Things (IoWST). The nodes were divided into three categories: sensor nodes, cluster heads, and coordinators. A consortium blockchain was established on the coordinators to store legitimate node identities and to enable the execution of smart contracts for authentication, data sharing, and non-repudiation among the sensor nodes. Ambient node data were stored using an AI-based interplanetary file system (IPFS).

Paper [27] offers a complete understanding of cybersecurity in WSNs, with a focus on modern machine learning (ML) and blockchain (BC) security approaches. It examines 171 recent studies on WSN security and investigates the incorporation of BC and ML into a lightweight security framework, with an emphasis on cyberattack detection and prevention within WSNs, as well as potential efficient BC and ML algorithms.

In [28], the authors proposed a new approach that combines linear network coding (LNC) with WSNs and blockchain-enabled IoT devices. This method was designed to reduce the energy consumption at each network node by applying LNC techniques. The authors conducted a thorough evaluation of the effectiveness and reliability of the model compared to other existing approaches. This study showed remarkable progress in several essential performance indicators, such as a larger number of active nodes, improved packet delivery

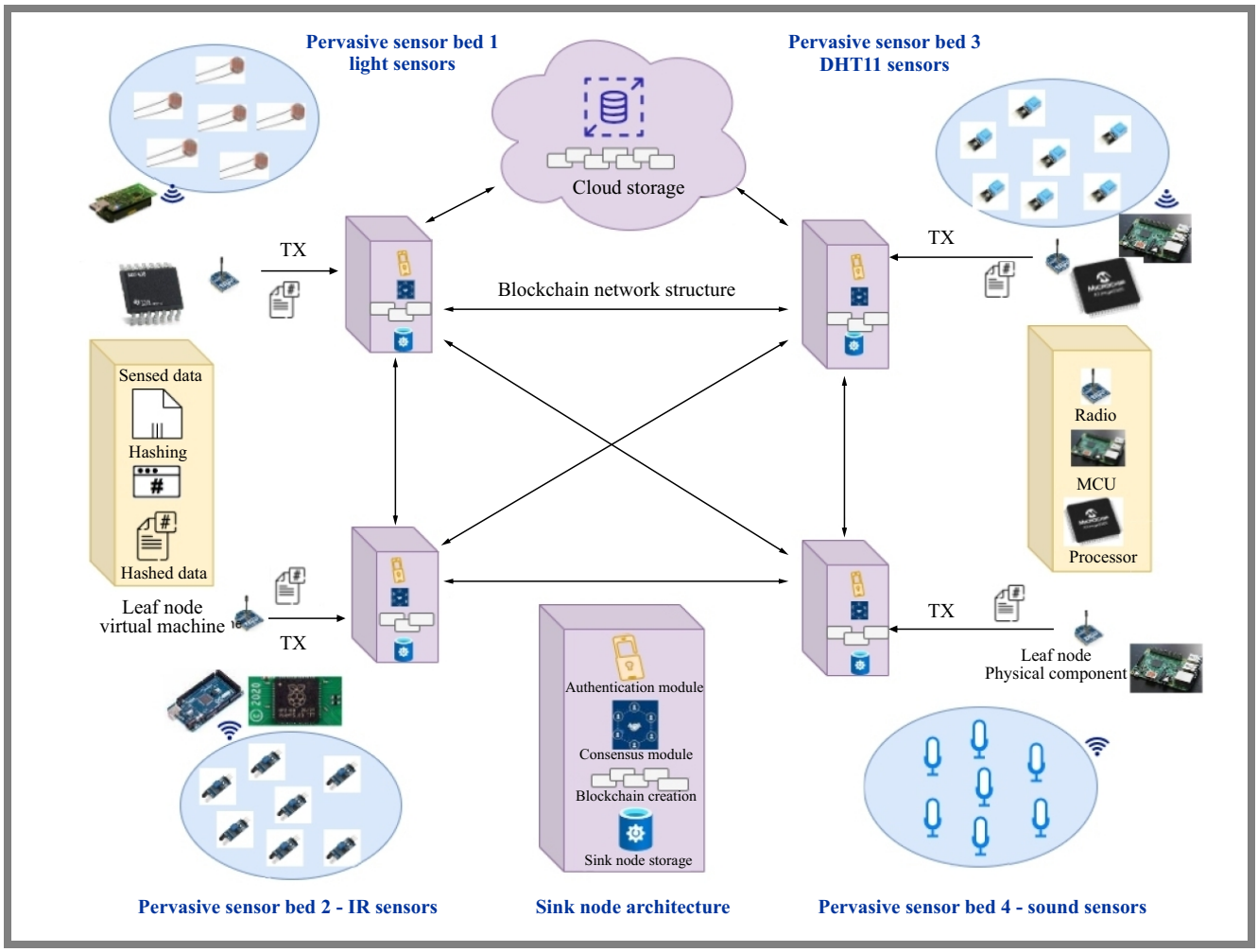

Fig. 1. System architecture (data integration using blockchain).

rate, increased throughput, and optimized remaining energy compared to current methods.

The authors of [29] proposed a new energy-efficient data collection mechanism (EEDAM) that uses the blockchain technology. This mechanism is designed to save energy resources by aggregating data at the cluster level. Edge computing is used to provide low-latency, trust-enhanced services to the IoT ecosystem. Blockchain integration is implemented in the cloud server to guarantee that the edge computing infrastructure is authenticated by the blockchain, thus providing a secure and reliable set of services to IoT devices.

The work described in [30] highlights the promising integration of IoT and BC in structural health monitoring, particularly for underground structures. The proposed blockchain-IoT network, with its locally centralized and globally decentralized features, offers a path toward more efficient, scalable, and secure SHM practices.

In summary, all papers mentioned above discussed the combination of BC with WSNs and IoT systems in the context of data management and energy efficiency. The results of these studies have been encouraging, suggesting that blockchain may be a beneficial solution to overcome difficulties and enhance the abilities of WSNs and IoT devices. Blockchain continues to open up the possibility of a more secure and reliable wireless sensing and IoT environment through data aggregation, security improvements, or efficient data management.

3. System Architecture

Figure 1 provides an overview of the architecture of the proposed model system. The network topology used in this model differs from that of conventional WSNs. On closer inspection, it is clear that the proposed topology follows a decentralized structure that includes sink nodes only. However, within the sensor zone, which extends up to the sink node, the conventional network topology used by PWSN is still in effect.

Figure 1 shows a system with four sensor zones, each connected to its own sink node. Every zone is equipped with multiple sensors, each with its own microcontroller and XBee radio technology, which form leaf nodes. These leaf nodes are connected directly to the sink nodes which are part of a decentralized network that is connected to a cloud environment with centralized cloud storage. Additionally, each sink node has local storage capabilities to efficiently manage data.

4. Methodology

The proposed method is structured into three distinct subsystems, each dedicated to a specific role: the data acquisition process at the leaf nodes and the subsequent creation of data blocks for the blockchain. The second subsystem deals with the reception of data blocks, verification of their authenticity, and their incorporation into the blockchain, while the third subsystem is responsible for disseminating the blockchain to cloud storage and local storage of sink nodes, ensuring its availability and redundancy.

4.1. Data Sensing and Data Block Creation

Different sensors are used in the WSN to capture signals emitted by the source devices. Each sensor has its own event and time ontology, allowing it to accurately determine its active state. This event and time ontology is incorporated into the virtual machine configuration of the sink node to reduce the amount of detected data, resulting in lower processing requirements, lower transmission overhead and thus reduced energy consumption. In this paper, the only CPU usage needed is for the generation of data blocks, which does not significantly affect energy consumption. Additionally, the sensed data blocks can be converted into the JSON format, reducing their size, and thus decreasing the energy needed for transmission. Algorithm 1 illustrates how data sensing, data block formation, and JSON data packet formatting are done in sequence.

The steps are described below:

- At each leaf node, the event and time ontology are preloaded, and the sensor is read in a specific time frame that is in line with the ontology.
- Leaf nodes sustain their sleep cycle effectively to minimize energy consumption.
- The leaf nodes collected the detected information and populated the values of the object generated from the SensorData class.
- The SensorData object is linked to the object created from the DataBlock object.
- Calculate the hash represented by Eq. (1) for SensorData using the SHA256 hash function.
- The data block is given the calculated hash as its current hash, whereas the previous hash is left blank for the sink node.
- The data block is then changed to the JSON format to decrease the packet size.
- The block is transmitted to the sink node for further processing.

4.2. Blockchain Formation and Data Integration

This work will exclude the consideration of energy requirements for sink nodes in WSNs, as they operate without relying on battery power. The sink nodes are established using a medium-range server configuration, incorporating a Python-designed virtual machine, and utilizing JSON files for storage.

Algorithm 1 Data block creation using hashing at leaf node

Require: Sensor devices, microcontroller
Ensure: DataBlock with generated hash sent to sink node
LeafVM $lvm \leftarrow$ LEAFVM(Microcontroller.devices)
SNode $ps \leftarrow$ PervasiveSensorNode(Sensor.devices)
empty SensorData object $sd \leftarrow$ Null
DataBlock $db \leftarrow$ DataBlock(lvm)
while lvm.devicePower() **and** lvm.isActive() **do**
  lvm.adaptiveDutyCycling()
  lvm.dataSampling()
  lvm.sleepScheduling()
  $sd \leftarrow$ SensorData($lvm.read(ps)$)
  db.sensorData $\leftarrow sd$
  db.previousHash $\leftarrow$ Null ▷ Assigned at sink node
  db.currentHash $\leftarrow db$.generateCurrentHash()
  ps.transmitToSinkNode(db)
end while
End

All the sink nodes, together with a central cloud storage system, will be configured using a decentralized network topology. The cloud storage system will also function as a node within this decentralized network. Blockchains will be stored in both the cloud storage system and the local storage of the sink nodes.

This proposed technique operates in a manner such that each sensor node in the network is not replicated to other sink nodes in the network. However, the information from the last hash will be shared among all the sink nodes, allowing the chain to be created by the information from the current hash, the previous hash, and the last hash of the DataBlock in a synchronized schedule, which reduces storage requirements by dividing it among the number of sink nodes. The diagram presented in Fig. 2 shows the functional block of this approach. Algorithm 2 describes the steps taken by the sink nodes to authenticate the data blocks, add them to the blockchain, notify other sink nodes, and store the block securely in the local sink nodes and last hash globally to all sink nodes and in the cloud.

The steps are described below:

- Receive data blocks from the leaf node in a JSON packet.
- Compute the hash represented by Eq. (1) for the data block, which is described in the DataBlock class and contains the sensor data specified in the SensorData class.
- Compare the hash from DataBlock to the hash that has been calculated.
- If the hash that is calculated and the hash that is compared are the same, it can be confirmed that the data block is genuine and has not been modified.
- Obtain the last hash from the blockchain from either a local storage or the network. Since the last hash is global to all sink nodes, the value will be the same throughout the network.
- Assign the last hash of the blockchain to the previous hash of the data block to be integrated.

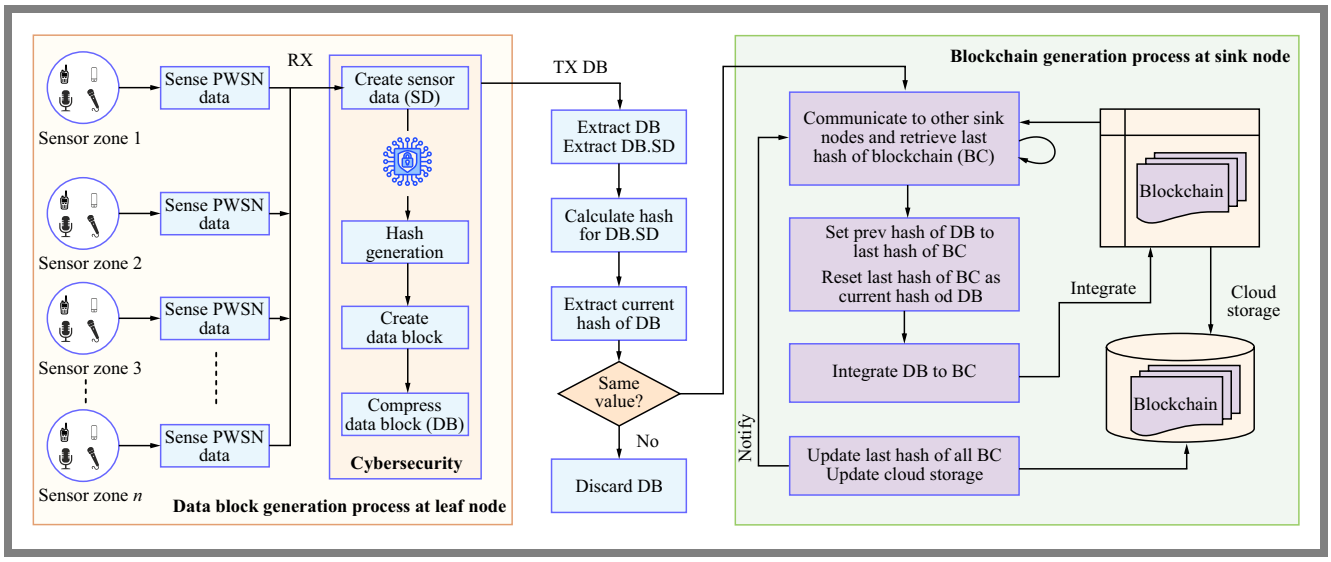

Fig. 2. Functional block diagram representing the blockchain at the sink node.

- Reset the last hash of the blockchain as the current hash of the data block.
- Notify and update the last hash of the blockchain of the entire network's sink node.
- Integrate the DataBlock to the "BlockChain"
- Store the DataBlock in the internal storage of the sink node.
- Store a copy of DataBlock to the cloud storage.
- The seven steps mentioned above must be coordinated with other sink nodes to guarantee that the integration of DataBlock to BlockChain and updating of the last hash of blockchain of the entire network are synchronized.

4.3. Transmission of Blockchain

The proposed model utilizes a standard cloud storage platform, such as Amazon S3 or Google Cloud, to maintain a backup of the blockchain core. Thus, even if a sink node fails, the sensor data remains accessible to users. The process $transmitCloudStorage()$, as described in Algorithm 2, demonstrates the transmission of the blockchain data to cloud storage. Furthermore, this approach includes web-based and mobile user interfaces to ensure user access to the sensor data.

4.4. Mathematical Model of the Proposed Architecture

The model captures energy consumption, blockchain latency, throughput, and data sensitivity using a set of well-defined performance metrics and equations.

Each sensor node collects environmental data, which is encapsulated in a data block. To ensure integrity and non-repudiation, a cryptographic hash is generated using the SHA256 algorithm:

$$\mathcal{H}(SD) = \text{SHA256}\big[\text{encode}(SD)\big] \,, \tag{1}$$

Algorithm 2 Blockchain creation and synchronization at sink node.

Require: Received DataBlock from leaf node
Ensure: Verified blockchain updated across all sink nodes

```
SinkVM svm ← SinkVM(SinkNode.devices)
local blockchain bc ← svm.getLocalBlockchain()
db ← svm.receiveBlock()
while svm.isActive() do
    if db = Null then
        db ← svm.receiveBlock()
    end if
    if bc = Null then
        db.previousHash ← Null
        bc.lastHash ← db.currentHash
        bc.bindFirstBlock(db)
    else
        bc.lastHash ← svm.getLastHashFromNetwork()
        if svm.authenticateBlock(db) then
            db.previousHash ← bc.lastHash
            bc.lastHash ← db.currentHash
            bc.appendBlock(db)
        end if
    end if
    svm.storeLocal(bc)
    svm.syncWithCloud(bc.lastHash)
    svm.notifyAllSinkNodes(bc.lastHash)
end while
```

where SD is the sensor data object:

$$SD = f(index, timestamp, sensedData, sensorId, location) \,. \tag{2}$$

The total energy consumed by a sensor node during sensing, processing, and transmission is given by:

$$E_{\text{node}} = \gamma \cdot \left[t_{\text{pac}} \cdot P_{\text{pac}} + t_{\text{sd}} \cdot P_{\text{sd}} + t_{\text{tx}} \cdot P_{\text{tx}} \right] , \tag{3}$$

where:

- $t_{\text{pac}}, t_{\text{sd}}, t_{\text{tx}}$ – time required for processor activation, data sensing, and data transmission,
- $P_{\text{pac}}, P_{\text{sd}}, P_{\text{tx}}$ – power consumption in watts for each respective operation,
- γ – adjustment factor based on the number of cycles or events.

Probability of data sensing:

$$P_{ds} = \frac{N_{sensed}}{N_{total}} \,. \tag{4}$$

Effective energy consumption:

$$E_{\text{eff}} = P_{ds} \cdot E_{\text{node}} \,. \tag{5}$$

Sensor node energy efficiency (SNEN):

$$\text{SNEN} = \left(1 - \frac{E_{\text{eff}}}{E_{\text{node}}}\right) \cdot 100\% \,. \tag{6}$$

Blockchain latency efficiency across sink nodes is modeled as follows:

$$\text{BCLE} = \frac{1}{n} \sum_{i=1}^{n} \frac{BT_i + TPT_i + NPT_i + TQT_i}{TTPT_i} \,, \tag{7}$$

where:

- BT – block time,
- TPT – transaction processing time,
- NPT – network propagation time,
- TQT – transaction queue time,
- $TTPT$ – total time to process a transaction.

Blockchain throughput efficiency (BCTE) evaluates the data handling capacity:

$$\text{BCTE} = \frac{1}{n} \sum_{i=1}^{n} \left(\frac{TPS_i}{TPT_i} \cdot (1 - BPR_i) \cdot (1 - LPR_i) \right) , \tag{8}$$

where:

- TPS – transactions per second,
- BPR – block processing rate,
- LPR – latency processing rate.

Pervasive data sensitivity (PDSN) measures the effectiveness of data compression and sensitivity across the sensor zones:

$$\text{PDSN} = \sum_{sZ}^{eZ} \sum_{t=t_{\text{in}}}^{t_d} \left(1 - \frac{sdSize\big(sEvent(t)\big)}{dbSize(t)} \right) \cdot 100\% \,. \tag{9}$$

Data block size:

$$dbSize(t) = P(sData) \cdot CR \cdot sdSize\big(sEvent(t)\big) \,, \tag{10}$$

where:

- $sdSize()$ – sensed data size,
- CR – compression ratio due to JSON encoding,
- $P(sData)$ – probability of sensing a relevant event.

5. Experimental Results and Performance Evaluation

The experiment integrates the sensed data using the proposed model, as well as the conventional PWSN model and actual blockchain models. The proposed hybrid offers good energy efficiency and its performance is measured by means of the sensor node energy efficiency (SNEN) metric with calculated energy efficiency. Equation (3) is used for computing the energy of the proposed model, whereas the probability of data sensing is expressed as Eq. (4).

Estimates of the consumed energy and energy efficiency are calculated using Eqs. (5) and (6), respectively. The efficiency of blockchain latency (BCLE) in PWSN can be affected by various elements that are exclusive to PWSN settings. This research uses Eq. (7) which takes into account the factors specified in Tab. 1. The present experiment evaluates the efficiency of blockchain throughput (BCTE) considering the factors listed in Tab. 1 and a corresponding equation has been formulated for the proposed model in Eq. (8). This research established that the effectiveness of data sensitivity (PDSN) for PWSN can be determined using Eq. (9), by means of which the decrease in data size (the size of events that occur in all sensor zones over a certain period of time) is calculated at some intervals, as is the size of the blockchain created by the proposed approach.

5.1. Experimental Setup

This research introduces three distinct sensor zones, each with its own set of sensor nodes. The first zone consists of six TelosB Mote (TPR2420) units, each with an MSP430 microcontroller as well as TPR2420 sensors for light, humidity, and temperature monitoring. The second zone has six Arduino Mega platforms, DHT11 sensors, and XBee radio modules for data transmission. The third zone consists of three Raspberry Pi boards, IR sensors, and XBee radio modules.

To facilitate data processing and transmission, each microcontroller at these sensor nodes will be integrated with a common virtual machine written in Python. This virtual machine will generate data blocks using hash functions and transmit them efficiently. Additionally, three high-end computers, each with a virtual machine of sink nodes written in Python, will serve as sink nodes for the three sensor zones to create a blockchain. The computers will store the blockchain locally in a JSON file, and the sink nodes will be interconnected via the Internet and linked to a cloud storage system hosted by Google Cloud for data storage and analysis.

Each experiment was carried out over a period of approximately 60 min of continuous operation. For every sensor zone, measurements were repeated ten times and the reported results represent mean values. Network traffic was evenly distributed among the sink nodes, with each handling 100 – 120 transactions per session. Latency and throughput were monitored in real time using the Python-based virtual machine module to ensure statistical consistency.

Tab. 1. Parameters considered to calculate performance metrics for the proposed model.

Symbol	Full form	Description
Parameters used in Eqs. (3), (4), and (6) to calculate SNEN		
E()	Energy function	Energy calculation in Joules
P()	Probability function	Probability of sensing event
t()	Time function	Time to process/transmit/receive
power()	Power function	Electrical power estimation in watts
Parameters used in Eq. (7) to calculate BCLE – blockchain latency efficiency		
BT	Block time	Average time to add new block to the blockchain
TPT	Transaction processing time	Time to process and validate transaction
NPT	Network propagation time	Time to take information about new block
TQT	Transaction queue time	Time to wait to be included in the block
TTP	Total time to process	Total time for confirmation on the blockchain
Parameters used in Eq. (8) to calculate BCTE – blockchain throughput efficiency		
TPS	Transaction per second	Number of transactions per second by sink node
TPT	Transaction processing time	Time to process and validate transaction
BPR	Block processing rate	Rate of addition of new blocks the blockchain
LPR	Latency processing rate	Rate of impact of latency within PWSN
Parameters used in Eq. (9) to calculate PDSN – pervasive data sensitivity		
sZ	Starting sensor zone	Index number for starting sensor zone
eZ	Ending sensor zone	Index number for ending sensor zone
t_d	Time duration	Total duration of entire PWSN
t_{in}	Sensing interval	Event sensing interval due to event semantic
sdSize()	Sensed event's data size	Function to measure data size
sEvent()	Sensed event	Event occurs at time t
dbSize()	Data block size	Function to measure data block for the event detected at time t
P(sData)	Probability of sensing data	Function to measure probability of event sensing
CR	Compression ratio	Compression ratio due to JSON data representation
RR	Reduction ratio	RR = P(sData) · CR

5.2. Experimental Results

The results of the experiment are evaluated using the setup described in Subsection 5.1 and Eqs. (3) to (8), respectively. Energy evaluations are conducted using the TelosB Mote, ATMEGA2560 MCU, XBee Pro, and Raspberry Pi platform data sheets to determine energy consumption. Table 2 presents the energy consumption at the leaf nodes in the sensor zone for single event sensing and data block formation with and without the hash. The average time required for detection, processing and transmission was calculated, and then the energy was determined in Joules. Figure 3 shows the data block created at the leaf node, along with the hash and the total time taken to generate the data block.

```
Data block details
Data block index: 1695780562
Data block size: 421 bytes
Sensor data size: 247 bytes
Hash size: 115 bytes
Data block index: 1695780562
Sensor ID: 10
Sensor zone ID: 1
Sensor lattitude: 22 25 58.5192
Sensor longitude: 87 51 35.5896
Sensor type: Temperature
Sensor value: 25.5
Hash:f87f9bb31f284b4169382679f7dd413f827a52a3f3cb67842b119b8
e8f2fe376
Processor active time: 0.18215274810791016 seconds
```

Fig. 3. Details of the data block generated at the leaf node.

Figure 4 shows the part of the blockchain generated by the entire network. It illustrates the structure of the DataBlock together with the corresponding SensorData. It reveals the present hash, the prior hash, and the last hash of the entire network. The last hash will be the same as the current hash if the synchronization functions correctly. The figure demonstrates

Tab. 2. Energy consumption (SD, TX, PAC) for a single DataBlock at the leaf node.

Energy consumption at leaf node (hash mode)				
Mode	Power [W]	Size [bytes]	Time [s]	Energy [J]
Data sensing (SD)	0.1925	247	0.0079	0.00152152
Data transmit (TX)	0.875	421	0.0134	0.011788
Processor active (PAC)	6.5	NA	0.201	1.3065
Total energy at leaf node				1.3198
Energy consumption at leaf node (no hash)				
Mode	Power [W]	Size [bytes]	Time [s]	Energy [J]
Data sensing (SD)	0.1925	211	0.0067	0.00129976
Data transmit (TX)	0.875	247	0.0079	0.006916
Average PAC	6.5	NA	0.171	1.1115
Total energy at leaf node				1.1197

that they are the same in all cases. If the last hash and the current hash do not match, this implies that the mismatched data block has not been included in the blockchain due to potential data manipulation.

Table 3 and Fig. 5 demonstrate the average energy consumption at the leaf nodes with different time intervals in various sensor zones. The energy is calculated by using a hash, without a hash, and with the use of the proposed hybrid model. aSNEN – see Eq. (6) – is also calculated for the proposed model using Tab. 3. The figure clearly indicates that a sing hash in PWSN is more energy-intensive than traditional PWSN, while the hybrid model proposed in this work significantly reduces the energy consumed in the sensor zones. The table shows that the energy efficiency for the sensor nodes (SNEN) is ≈40%.

Table 4 provides the blockchain latency (BCLE) for all sink nodes involved in the PWSN using Eq. (7), together with the average block creation time (BT), transaction processing time (TPT), network processing time (NPT), transaction queue time for the specified time intervals and total number of events. The total time taken to process a blockchain is also calculated. The proposed hybrid technique yields an average BCLE of

Tab. 3. Energy consumption using hash and percentage of SNEN.

Time	Inter-val	Total event	P(sEvent)	Energy using hash	Energy hy-brid	SNEN
200	4	50	0.61	65.99	40.25	39%
400	5	80	0.59	105.58	62.29	41%
600	4	150	0.67	197.97	132.64	33%
800	5	160	0.52	211.17	109.81	48%
1000	4	250	0.52	329.95	171.57	48%

Fig. 4. Blockchain generated by the proposed model for PWSN.

80%, which is significantly better than in a scenario in which blockchain is used with the traditional PWSN model.

Equation (8) is used to calculate the average blockchain throughput (BCTE) for all sink nodes in the network, i.e. the transactions per second (TPS) of all sink nodes. Then, the total processing time (TPT) and blockchain processing time (BPT) are calculated and the average latency rate (LPR) calculated in Tab. 5 is used. The average BCTE is ≈37% for the proposed hybrid model, which is also significantly better than when using blockchain with the traditional PWSN model.

Equation (9) determines the prevalence of data sensitivity shown in Tab. 6, indicating the percentage of data reduction achieved by the proposed technique. This calculation is based on the actual data size of events that occurred across the network over a period of time, as well as the actual blockchain

Tab. 4. Blockchain latency efficiency (BCLE) for the whole network

Time	Total event	BT [s]	TPT [s]	NPT [s]	TQT [s]	TTP [s]	BCLE
200	50	2.65	1.05	10.05	1.55	12.67	82.84%
400	80	4.40	2.16	16.4	2.32	21.25	84.08%
600	150	8.10	4.20	29.85	3.60	40.20	87.86%
800	160	8.16	3.84	34.08	4.16	42.08	83.75%
1000	250	14.00	5.75	52.75	6.75	67.00	84.54%

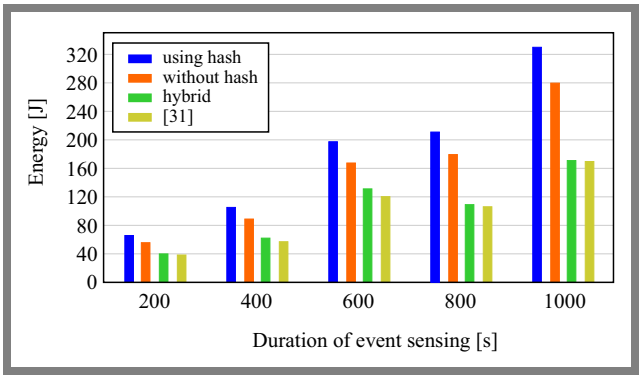

Fig. 5. Sensor zone energy comparison with [31].

size, which includes data blocks. The results suggest that the proposed technique is capable of reducing the size of the data approximately by 30%.

5.3. Comparison and Performance Analysis

This study focuses mainly on evaluating the energy efficiency within the sensor zone, particularly with respect to battery-operated leaf nodes. Although most of the research related to data integration using blockchain in ubiquitous wireless sensor networks tends to emphasize aspects such as integrity, security, and overall network life expectancy, this work takes a distinctive approach by directly comparing its energy efficiency within the sensor zone with the technique introduced in [31]. Figure 5 presents a comparison of the average energy consumption of the leaf nodes within the sensor zones in four different modes: with hash use, without hash use, using the proposed hybrid model and using the model introduced in [31].

Despite the fact that the proposed model includes hash usage for the creation of data blocks, which is known to require more energy for data processing and transmission, the figure shows that it consumes less energy than the scenarios of using hash exclusively and adhering to traditional methods, i.e. [31]. This is further supported by the comparison of energy efficiency in Fig. 6, calculated on the basis of Eq. (6) and the data represented in Tab. 3, which shows that the proposed model has an efficiency approximately 10% higher than [31].

Tab. 5. Blockchain throughput efficiency (BCTE) for the entire network.

Time	Interval	TPS	TPT [s]	BPR [s]	LPR	BCTE [%]
200	4	0.25	0.084	0.053	0.823	49.88
400	5	0.20	0.105	0.055	0.841	28.62
600	4	0.25	0.084	0.054	0.879	34.06
800	5	0.20	0.105	0.051	0.838	29.28
1000	4	0.25	0.084	0.056	0.845	43.54

Tab. 6. Pervasive data sensitivity (PDSN) for the whole network.

Duration	Event data size [bytes]	RR factor	Data block size [bytes]	PDSN
200	29640	0.53	19995	32.53%
400	74100	0.59	55648	24.90%
600	88920	0.54	61119	31.26%
800	148200	0.57	107524	27.44%
1000	148200	0.58	109411	26.17%

In order to fully evaluate the performance of the proposed hybrid model, this work draws a comparison with the technique introduced in [16], which also aims to improve the security and management of IoT data through a decentralized blockchain-based architecture.

To compute BCLE and BCTE for the approach from [16], this work relies on the data provided in their figures, and for the proposed model, this work uses Eqs. (7) and (8). The resulting BCLE values are presented in Tab. 4, and the BCTE values are detailed in Tab. 5. Based on the calculations and analysis, this work compiles the findings into a comparative graph shown in Fig. 7. In particular, the BCLE values for both models are quite similar. However, the proposed model exhibits a BCTE that is ≈25% better than that of the model proposed in [16].

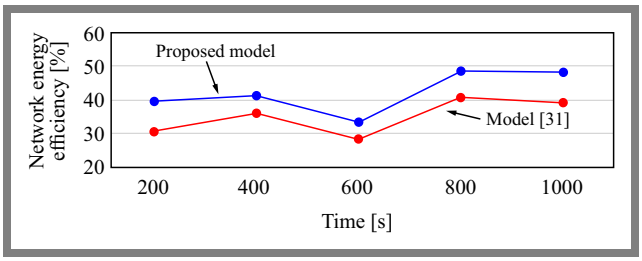

Fig. 6. Comparison of network energy efficiency.

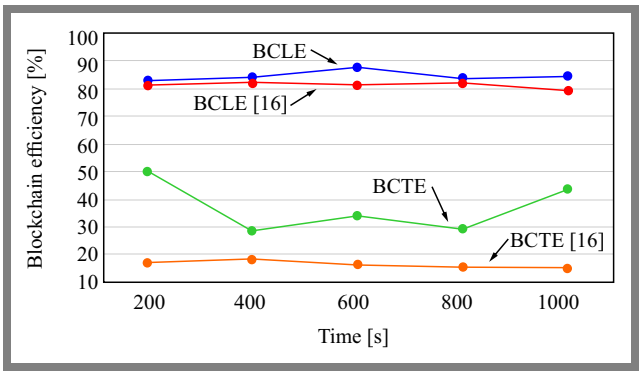

Fig. 7. Comparison of BCLE and BCTE metrics for the proposed model with [16].

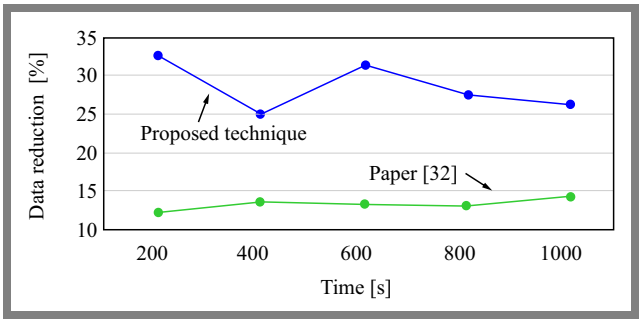

Fig. 8. Comparison of the data reduction percentage with [32].

The proposed technique utilizes the blockchain technology, increasing the amount of transmitted data. To address this problem, the proposed technique introduces a semantic mechanism to reduce the volume of data and creates a data block using the JSON format. The authors of [32] utilize semantic rules to reduce the weight of sensor data in different environments. Figure 8 shows that the proposed technique offers a significant improvement in terms of the data reduction percentage (≈20%) compared to [32].

6. Conclusion and Future Scope

This paper presented a novel hybrid architecture for pervasive wireless sensor networks (PWSNs) that integrates the blockchain technology to achieve secure and energy-efficient IoT data processing. The proposed design combines decentralized sink-node blockchain management with centralized sensing zones, providing both security and energy efficiency.

Implementation using TelosB Mote and Raspberry Pi devices demonstrated real-time data integration, while experimental results validated the model's higher energy efficiency, reduced latency, and improved throughput compared to existing methods.

Future studies will extend this prototype by incorporating consensus mechanisms such as Proof of Work (PoW) and Proof of Stake (PoS) to analyze their impact on energy consumption and latency, thus improving the robustness and adaptability of blockchain-enabled PWSNs.

References

[1] D. Berdik *et al.*, "A Survey on Blockchain for Information Systems Management and Security", *Information Processing and Management*, vol. 58, art. no. 102397, 2021 (https://doi.org/10.1016/j.ipm.2020.102397).

[2] S. Darla and C. Naveena, "Survey on Securing Internet of Things through Blockchain Technology", *2022 International Conference on Electronics and Renewable Systems (ICEARS)*, Tuticorin, India, 2022 (https://doi.org/10.1109/ICEARS53579.2022.9752316).

[3] T.K. Araghi, D. Megías, and A. Rosales, "Evaluation and Analysis of Reversible Watermarking Techniques in WSN for Secure, Lightweight Design of IoT Applications: A Survey", *Advances in Information and Communication*, vol. 652, pp. 695–708, 2023 (https://doi.org/10.1007/978-3-031-28073-3_47).

[4] S. Kumar and V. Singh, "A Review of Digital Signature and Hash Function Based Approach for Secure Routing in VANET", *2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS)*, Coimbatore, India, 2021 (https://doi.org/10.1109/ICAIS50930.2021.9395882).

[5] A.A. Monrat, O. Schelän, and K. Andersson, "A Survey of Blockchain from the Perspectives of Applications, Challenges, and Opportunities", *IEEE Access*, vol. 7, pp. 117134–117151, 2019 (https://doi.org/10.1109/ACCESS.2019.2936094).

[6] A.K. Sharma and S. Mittal, "Cryptography and Network Security Hash Function Applications, Attacks and Advances: A Review", *2019 Third International Conference on Inventive Systems and Control (ICISC)*, Coimbatore, India, 2019 (https://doi.org/10.1109/ICISC44355.2019.9036448).

[7] A. Singh and S. Gupta, "Learning to Hash: A Comprehensive Survey of Deep Learning-based Hashing Methods", *Knowledge and Information Systems*, vol. 64, pp. 2565–2597, 2022 (https://doi.org/10.1007/s10115-022-01734-0).

[8] S. Das and U. Mondal, "Acoustic Data Acquisition and Integration for Semantic Organization of Sentimental Data and Analysis in a PWSN", *Multimedia Tools and Applications*, vol. 84, pp. 26755–26777, 2024 (https://doi.org/10.1007/s11042-024-20229-4).

[9] S. Das and U. Mondal, "Energy Efficient Acoustic Sensor Data Integration in Hybrid Mode Operated Pervasive Wireless Sensor Network", *Telecommunication Systems*, vol. 87, pp. 61–72, 2024 (https://doi.org/10.1007/s11235-024-01165-y).

[10] S. Das and U. Mondal, "Pilot Agent Implied Efficient Data Communication in Pervasive Acoustic Wireless Sensor Network", *Telecommunication Systems*, vol. 88, art. no. 50, 2025 (https://doi.org/10.1007/s11235-025-01281-3).

[11] H. Guo and X. Yu, "A Survey on Blockchain Technology and its Security", *Blockchain: Research and Applications*, vol. 3, 2022 (https://doi.org/10.1016/j.bcra.2022.100067).

[12] C.V. Nguyen *et al.*, "Blockchain technology in wireless sensor network: benefits and challenges", *ICSES Transactions on Computer Networks and Communications*, vol. 10, pp. 1–4, 2021.

[13] S. Hsiao and W. Sung, "Utilizing Blockchain Technology to Improve WSN Security for Sensor Data Transmission", *Computers, Materials & Continua*, vol. 68, pp. 1899–1918, 2021 (https://doi.org/10.32604/cmc.2021.015762).

[14] T. Huynh-The *et al.*, "Blockchain for the Metaverse: A Review", *Future Generation Computer Systems*, vol. 143, pp. 401–419, 2023 (https://doi.org/10.1016/j.future.2023.02.008).

[15] S. Hsiao and W. Sung, "Employing Blockchain Technology to Strengthen Security of Wireless Sensor Networks", *IEEE Access*, vol. 9, pp. 72326–72341, 2021 (https://doi.org/10.1109/ACCESS.2021.3079708).

[16] A. Maftei, A. Lavric, A. Petrariu, and V. Popa, "Massive Data Storage Solution for IoT Devices Using Blockchain Technologies", *Sensors*, vol. 23, art. no. 1570, 2023 (https://doi.org/10.3390/s23031570).

[17] L.K. Ramasamy *et al.*, "Blockchain-based Wireless Sensor Networks for Malicious Node Detection: A Survey", *IEEE Access*, vol. 9, pp. 128765–128785, 2021 (https://doi.org/10.1109/ACCESS.2021.3111923).

[18] B. Sudheer and K. Sujatha, "A Brief Survey on Data Aggregation and Data Compression Models Using Blockchain Model in Wireless Sensor Network", *2023 International Conference on Innovative Data Communication Technologies and Application (ICIDCA)*, Uttarakhand, India, 2023 (https://doi.org/10.1109/ICIDCA56705.2023.10100009).

[19] O. Khalaf and G. Abdulsahib, "Optimized Dynamic Storage of Data (ODSD) in IoT Based on Blockchain for Wireless Sensor Networks", *Peer-to-Peer Networking and Applications*, vol. 14, pp. 2858–2873, 2021 (https://doi.org/10.1007/s12083-021-01115-4).

[20] M. Madhi, A. Al-Bakry, and A. Farhan, "IoT Conception Based on Blockchain Technology: A Review", *Al-Mansour Journal*, vol. 39, pp. 1–9, 2023 (https://muc.edu.iq/oldwebsite/mucj/39/english/e4-b39.pdf).

[21] Y. Ren *et al.*, "Incentive Mechanism of Data Storage Based on Blockchain for Wireless Sensor Networks", *Mobile Information Systems*, 2018 (https://doi.org/10.1155/2018/6874158).

[22] M. Rajhi and A. Hakami, "A Cryptographic Iterative Hash Function Scheme for Wireless Sensor Network (WSNs) Security Enhancement for Sensor Data Transmission in Blockchain", *TechRxiv*, 2022 (https://doi.org/10.36227/techrxiv.19323308.v1).

[23] Z. Cui *et al.*, "A Hybrid Blockchain-based Identity Authentication Scheme for multi-WSN", *IEEE Transactions on Services Computing*, vol. 13, pp. 241–251, 2020 (https://doi.org/10.1109/TSC.2020.2964537).

[24] A. Mubarakali, "An Efficient Authentication Scheme Using Blockchain Technology for Wireless Sensor Networks", *Wireless Personal Communications*, vol. 127, pp. 255–269, 2021 (https://doi.org/10.1007/s11277-021-08212-w).

[25] R. Goyat *et al.*, "Blockchain-based Data Storage with Privacy and Authentication in Internet of Things", *IEEE Internet of Things Journal*, vol. 9, pp. 14203–14215, 2020 (https://doi.org/10.1109/JIOT.2020.3019074).

[26] A. Khan, N. Javaid, M. Khan, and I. Ullah, "A Blockchain Scheme for Authentication, Data Sharing and Nonrepudiation to Secure Internet of Wireless Sensor Things", *Cluster Computing*, vol. 26, pp. 945–960, 2023 (https://doi.org/10.1007/s10586-022-03722-z).

[27] S. Ismail, D. Dawoud, and H. Reza, "Securing Wireless Sensor Networks Using Machine Learning and Blockchain: A Review", *Future Internet*, vol. 15, art. no. 200, 2023 (https://doi.org/10.3390/fi15060200).

[28] N. Alghamdi and M. Khan, "Energy-efficient and Blockchain Enabled Model for Internet of Things (IoT) in Smart Cities", *Computers, Materials and Continua*, vol. 66, pp. 2509–2524, 2021 (https://doi.org/10.32604/cmc.2021.014180).

[29] A. Ahmed *et al.*, "An Energy-efficient Data Aggregation Mechanism for IoT Secured by Blockchain", *IEEE Access*, vol. 10, pp. 11404–11419, 2022 (https://doi.org/10.1109/ACCESS.2022.3146295).

[30] B. Jo, R. Khan, and Y. Lee, "Hybrid Blockchain and Internet-of-Things Network for Underground Structure Health Monitoring", *Sensors*, vol. 18, art. no. 4268, 2018 (https://doi.org/10.3390/s18124268).

[31] M.S. Andhare *et al.*, "Design and Implementation of Wireless Sensor Network for Environmental Monitoring", *International Journal of Health Sciences*, vol. 6, pp. 3158–3169, 2022 (https://doi.org/10.53730/ijhs.v6ns4.9085).

[32] G. Urkude and M. Pandey, "Contextual Triple Inference Using a Semantic Reasoner Rule to Reduce the Weight of Semantically Annotated Data on Fail-safe Gateway for WSN", *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, pp. 5107–5121, 2021 (https://doi.org/10.1007/s12652-020-02836-9).

Sushovan Das, M.Tech.
Department of CSE
https://orcid.org/0000-0003-2759-3902
E-mail: das.sushovan@gmail.com
College of Engineering and Management, Kolaghat, Purba Medinipur, India
https://www.cemkolaghat.in

Uttam Kr. Mondal, Ph.D.
Department of Computer Science
https://orcid.org/0000-0002-7807-3002
E-mail: uttam_ku_82@yahoo.co.in
Vidyasagar University, Midnapur, India
https://www.vidyasagar.ac.in

Babai-guided Interference-aware Adaptive QRD-M Detection in MIMO-OFDM Communication Systems

Mar Mar Lwin and Mohd Fadzli Mohd Salleh

Universiti Sains Malaysia, Nibong Tebal, Malaysia

https://doi.org/10.26636/jtit.2025.4.2290

Abstract — This paper presents an adaptive QRD-M detection algorithm designed to reduce the computational complexity of MIMO systems while maintaining near-maximum likelihood detection (near-MLD) performance. The proposed method introduces a dynamic threshold mechanism based on a breadth-first tree search, where pruning is guided by both symbol reliability and interlayer interference derived from the upper-triangular structure of the QR-decomposed channel matrix. The threshold is further refined using a Babai estimate obtained from Lenstra–Lenstra–Lovász (LLL) lattice reduction, allowing the algorithm to adaptively adjust the candidate set at each detection stage. The simulation results across 4 × 4 and 8 × 8 MIMO systems using 16-QAM and 64-QAM modulation schemes demonstrate that the proposed Babai-guided interference-aware adaptive QRD-M (BIA-QRD-M) algorithm achieves near-MLD performance. The proposed method achieves a reduction of up to 49% in the average number of branch metric computations at high SNR and an approximately 29% reduction over the entire 0 – 25 dB SNR range, compared to conventional QRD-M in an 8 × 8 MIMO-OFDM system with 16-QAM modulation.

Keywords — LLL lattice reduction, MIMO-OFDM systems, QRD-M detection

1. Introduction

Multiple input, multiple output (MIMO) systems are a key technology of high-capacity wireless communication solutions, offering substantial gains in spectral efficiency and link reliability. However, the associated symbol detection task becomes increasingly complex with high-order modulation and large antenna configurations. While maximum likelihood detection (MLD) [1] achieves optimal performance, it suffers from exponential grooving computational complexity, making it impractical for real-time implementations in most scenarios.

By contrast, linear detectors offer low implementation complexity but suffer from performance degradation under ill-conditioned channels. Lattice reduction preprocessing can partially mitigate this drawback [2]. To overcome this problem, a variety of suboptimal detection algorithms have been developed to approximate MLD with reduced complexity. Notable examples include sphere decoding (SD) [3], which performs an efficient search within a hypersphere, and the QRD-M algorithm [4], which limits the number of candidate paths retained during detection.

These methods aim to strike a balance between detection performance and computational feasibility, forming a basis for ongoing research into adaptive and low-complexity MIMO detection techniques.

Among these approaches, the QR decomposition with M algorithm (QRD-M) has gained popularity due to its structured tree search and consistent near-MLD performance. It first applies QR decomposition to the channel matrix and then selects M most reliable candidates at each detection layer based on the accumulated Euclidean distance. However, the algorithm still incurs high and fixed complexity, particularly when a large M is needed to maintain accuracy under high SNR or high-order modulation.

This paper addresses the performance-complexity trade-off in QRD-M, i.e. maintains near MLD performance while reducing the number of branch metric computations, using a Babai-guided, interference-aware adaptive threshold that scales with SNR.

2. Related Works

To address the computational burden of QRD-M, various improvements have been proposed. In [5], bounding techniques were applied to constrain the search region and reduce average complexity without significant performance degradation. In [6], an adaptive threshold was introduced in the K-best sphere decoding, dynamically adjusting the candidate list according to channel conditions. Within the QRD-M framework, path elimination is pruned. This approach offers complexity savings that are inherently dependent on the modulation method proposed in [7], where branches with accumulated Euclidean distances exceeding the minimum at each layer are of order.

In [8], a Babai-based thresholding approach was proposed, where the candidate set is adaptively expanded when initial pruning yields too few surviving paths. This method achieves lower average complexity in many scenarios by combining aggressive pruning with selective recovery of candidates. However, its dynamic expansion behavior can introduce vari-

ability in computational load and requires careful threshold tuning.

In [9], the set was restricted to a modulation-specific neighborhood centered around a QR-based estimate. While this approach effectively reduces complexity, it lacks adaptability across different modulation formats and does not account for inter-layer interference or symbol reliability. The radius of the neighborhood is manually configured per modulation format, limiting the flexibility in heterogeneous or dynamically varying scenarios.

These limitations highlight the need for a more general pruning strategy that takes advantage of standard preprocessing, avoids heuristic dependencies, and adapts reliably to varying MIMO-OFDM configurations. Such trade-offs described across prior works motivate the development of a pruning method that offers adaptively bounded complexity across SNR regimes and system configurations, without reliance on modulation-specific thresholds.

Despite these efforts, achieving near-MLD performance with low complexity and without relying on modulation-specific structures or heuristics remains an open challenge. This paper proposes an adaptive pruning strategy based on interference-aware thresholding, which dynamically adjusts the candidate set using symbol reliability and channel structure, and does so without relying on modulation-specific heuristics.

Complementary research investigates iterative detectors aided by lattice reduction [10] and model driven deep learning detectors [11]. These approaches typically require soft information exchange or offline training, whereas the present study advances the hard decision QRD-M family with a training-free, rule-based threshold that directly reduces branch metric (SED) counts.

Here, a Babai-guided, interference-aware adaptive threshold with SNR dependent scaling is introduced for QRD-M to adjust the survivor set per layer without training or soft output. The algorithm integrates the deviation from the Babai point with a normalized interlayer interference term derived from the upper triangular matrix, thereby stabilizing pruning across SNR regimes and channel conditions. The resulting detector reduces branch metric (SED) counts while preserving near MLD performance, and is validated on 4 × 4 and 8 × 8 MIMO-OFDM with 16-QAM and 64-QAM under flat and frequency selective channels.

3. System Description

A spatial multiplexing multiple input multiple output (MIMO) system is modeled with N_T transmit antennas and N_R receive antennas, under the assumption that $N_R \geqslant N_T$. The received signal vector $\boldsymbol{y} \in \mathbb{C}^{N_R}$ is given by [12]:

$$\boldsymbol{y} = \mathbf{H}\boldsymbol{x} + \boldsymbol{n}\,, \tag{1}$$

where $\mathbf{H} \in \mathbb{C}^{N_R \times N_T}$ denotes the complex-valued flat fading channel matrix, the transmitted symbol vector $\boldsymbol{x} \in S^{N_T}$ is drawn from a constellation set S, such as QAM or PAM, and $\boldsymbol{n} \sim \mathcal{CN}(0, \sigma^2 \mathbf{I}_{N_R})$ is additive white Gaussian noise with variance σ^2.

To enable efficient detection, QR decomposition is applied to channel matrix $\mathbf{H}$, yielding the following:

$$\mathbf{H} = \mathbf{QR}\,, \tag{2}$$

where $\mathbf{Q} \in \mathbb{C}^{N_R \times N_T}$ is a unitary matrix (i.e. $\mathbf{Q}^H\mathbf{Q} = \mathbf{I}_{N_T}$) and $\mathbf{R} \in \mathbb{C}^{N_T \times N_T}$ is an upper-triangular matrix.

Multiplying both sides of Eq. (1) by $\mathbf{Q}^H$ results in an upper-triangular form:

$$\hat{\boldsymbol{y}} = \mathbf{Q}^H\boldsymbol{y} = \mathbf{R}\boldsymbol{x} + \hat{\boldsymbol{n}}\,. \tag{3}$$

Here, $\hat{\boldsymbol{y}} \in \mathbb{C}^{N_R}$ is the transformed received vector and $\hat{\boldsymbol{n}} = \mathbf{Q}^H\boldsymbol{n}$ retains the same statistical properties due to the unitary nature of $\mathbf{Q}$.

Based on the triangular system model, the goal of MIMO detection is to estimate the vector of the transmitted symbol vector $\boldsymbol{x} \in S^{N_T}$ from the transformed observation $\hat{\boldsymbol{y}} = \mathbf{Q}^H\boldsymbol{y}$. This is achieved by finding the vector that minimizes the discrepancy between the received signal and its reconstruction through the channel. Mathematically, the detection task is formulated as an integer least squares (ILS) problem given by:

$$\hat{\boldsymbol{x}} = \arg \min_{\boldsymbol{x} \in S^{N_T}} \|\hat{\boldsymbol{y}} - \mathbf{R}\boldsymbol{x}\|^2\,, \tag{4}$$

where $\mathbf{R}$ is the upper-triangular matrix obtained from QR decomposition. Solution $\hat{\boldsymbol{x}}$ represents the closest point in the lattice generated by $\mathbf{R}$ to observation $\hat{\boldsymbol{y}}$, under the constraint that the components of $\boldsymbol{x}$ are drawn from a discrete modulation set S.

Solving this problem exactly yields the maximum likelihood (ML) estimate, but its computational complexity grows exponentially with N_T and the constellation size. Therefore, suboptimal but efficient detection algorithms such as SD and QRD-M are typically used to approximate the ML solution.

3.1. QRD-M Detection

QRD-M is a breadth-first tree search algorithm designed to approximate the solution of the integer least squares (ILS) problem defined in Eq. (4), based on the representation of the triangular system in Eq. (3). The detection objective is to find that the transmitted vector $\boldsymbol{x} \in S^{N_T}$ minimizes the squared Euclidean distance between the transformed received signal $\hat{\boldsymbol{y}}$ and its reconstruction via the upper-triangular matrix $\mathbf{R}$.

The squared Euclidean distance (SED) is given by:

$$\text{SED} = \|\hat{\boldsymbol{y}} - \mathbf{R}\boldsymbol{x}\|^2\,. \tag{5}$$

Because $\mathbf{R}$ is upper-triangular, each component of the residual depends only on $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_{N_T}$. Hence Eq. (5) can be written as:

$$\|\hat{\boldsymbol{y}} - \mathbf{R}\boldsymbol{x}\|^2 = \sum_{i=1}^{N_T} \left\| \hat{\boldsymbol{y}}_i - \sum_{j=i}^{N_T} \mathbf{R}_{ij}\boldsymbol{x}_j \right\|^2\,, \tag{6}$$

where $\mathbf{R}_{ij}$ denotes the element in the i-th row and j-th column of matrix $\mathbf{R}$, $\hat{\boldsymbol{y}}_i$ is the i-th received signal after nulling and $\boldsymbol{x}_j$ is the j-th transmit signal.

This layer-wise expansion reveals the interference structure embedded in each received component $\hat{\boldsymbol{y}}_i$. Specifically, the term $\sum_{j=i}^{N_T} \mathbf{R}_{ij}\, \boldsymbol{x}_j$ represents inter-layer interference from symbols $\boldsymbol{x}_j$ not yet decided in the detection process. As detection progresses from stage $i = N_T$ (root node) to stage $i = 1$ (first layer), the interference accumulates and becomes more significant, particularly in ill-conditioned channels.

The conventional QRD-M algorithm mitigates this by maintaining a fixed number M of candidate paths at each stage, selecting the highest M symbol extensions with the smallest partial Euclidean distance. Although conventional QRD-M provides a computationally efficient approximation to maximum likelihood detection, its fixed candidate size does not respond to variations in interference or noise conditions. This limitation motivates the development of an adaptive approach.

3.2. Proposed Method

The proposed Babai-guided interference-aware adaptive QRD-M (BIA-QRD-M) algorithm enhances the conventional QRD-M by adaptively pruning symbol candidates at each detection layer based on the local structure of the received signal. Unlike fixed m QRD-M, which retains a constant number of candidates regardless of channel or noise conditions, BIA-QRD-M dynamically adjusts the pruning threshold using a combination of Babai point deviation and a normalized interference plus noise term, both computed from the QR decomposition of the LLL-reduced basis channel matrix. This integration of lattice reduction enhances orthogonality and improves the reliability of the Babai estimate used for adaptive pruning.

The QRD-M algorithm improves detection accuracy over linear detectors by exploring multiple symbol candidates at each detection layer. However, the fixed-M QRD-M expands a predetermined number of candidates regardless of signal quality or interference level, resulting in either excessive complexity or insufficient accuracy.

To address this, an adaptive pruning strategy is proposed that dynamically adjusts the number of candidates in each layer based on symbol reliability and the magnitude. At each detection layer i, the pruning threshold η_i determines the allowable deviation from the Babai estimate, taking into account both symbol reliability and the impact of interlayer interference.

The threshold at layer i is defined as:

$$\eta_i = \frac{\gamma \cdot |\hat{\boldsymbol{b}}_i - \widetilde{\boldsymbol{y}}_i| + \delta \cdot \left(\frac{\sum_{j=i+1}^{N_T} \left|\widetilde{\mathbf{R}}_{i,j}\right|}{\left|\widetilde{\mathbf{R}}_{i,i}\right|} \right)}{1 + \alpha \cdot SNR_{\text{linear}}}, \tag{7}$$

where:

- $\widetilde{\mathbf{Q}}$ and $\widetilde{\mathbf{R}}$ are LLL-reduced basis obtained from the complex-valued channel matrix $\mathbf{H}$, while $\widetilde{\mathbf{R}}_{i,j}$ are entries of the upper-triangular matrix $\widetilde{\mathbf{R}}$,
- $\hat{\boldsymbol{b}}_i$ is the Babai estimate at layer i, calculated from the ZF solution $\hat{\boldsymbol{x}}_{ZF} = \widetilde{\mathbf{R}}^{-1}\hat{\boldsymbol{b}}_i$, followed by nearest neighbour rounding,
- $\widetilde{\boldsymbol{y}}_i = [\widetilde{\mathbf{Q}}^H \boldsymbol{y}]_i$ is the i-th component of the transformed received vector,
- $\gamma > 0$ and $\delta > 0$ are user-defined parameters,
- $\alpha \geqslant 0$ is a scaling parameter that adjusts the overall pruning aggressiveness with respect to SNR, ensuring that the threshold becomes tighter at high SNR and looser at low SNR,
- $SNR_{\text{linear}} = 10^{\frac{SNR_{dB}}{10}}$ converts the SNR value from decibels to a linear scale.

For all simulations, $\alpha = 0.5$ is used for a flat-fading channel, while $\alpha = 0.02$ is applied to frequency-selective fading channels. Parameters γ and δ are fixed at 1.5 and 2.0, respectively, in all scenarios.

In contrast to fixed thresholds based on Babai or neighborhood margins, the proposed scheme determines the pruning level from the deviation to the Babai point, together with a normalized measure of interlayer interference computed from the upper-triangular factor of the QR decomposition, with explicit SNR dependent scaling. This rule-based, layer adaptive mechanism reduces branch metric (SED) counts while maintaining near-MLD candidates.

This formulation incorporates three key observations:

1. **Symbol reliability.** The term $|\hat{\boldsymbol{b}}_i - \widetilde{\boldsymbol{y}}_i|$ quantifies the deviation between the Babai estimate and the transformed received symbol in the LLL-reduced domain. A smaller value indicates that the Babai estimate closely aligns with the underlying received symbol, suggesting high confidence in rounding decision and allowing for tighter pruning.

2. **Normalized interference term.** The summation $\sum_{j=i+1}^{N_T} \left|\widetilde{\mathbf{R}}_{i,j}\right|$ captures the cumulative effect of undecided symbols from lower layers. Dividing this by the diagonal term $\left|\widetilde{\mathbf{R}}_{i,i}\right|$ normalizes the interference with respect to the signal strength at the current layer. A higher normalized value implies stronger residual interference, prompting looser pruning to maintain detection robustness.

3. **SNR-dependent scaling.** The denominator term $1 + \alpha \cdot SNR_{\text{linear}}$ introduces a global control mechanism that tightens the threshold as the signal-to-noise ratio increases. At high SNR, where symbol estimates become more reliable, the threshold becomes smaller, enabling more aggressive pruning. At low SNR, the threshold is relaxed, ensuring robustness under noise-dominant conditions.

Together, these three components allow the threshold to dynamically adapt based on both local layer conditions (symbol confidence and interference) and global channel reliability (SNR). This adaptive mechanism balances detection performance and computational complexity more effectively than fixed-M approaches.

Parameters γ, δ, and α serve as tuning knobs operating in the following manner:

- Increasing γ makes the pruning more sensitive to the Babai point error, emphasizing symbol reliability,
- Increasing δ gives more weight to normalized interference, relaxing the threshold in high interference layers,
- Increasing α intensifies the influence of SNR, imposing a tighter threshold as channel conditions improve.

This design ensures that a larger number of candidates is evaluated only when necessary, achieving near-optimal detection performance while significantly reducing average complexity across a wide range of SNR and channel conditions.

The detection procedure is as follows:

Initialization. The detection process begins by applying the QR decomposition to the channel matrix and transforming the received vector accordingly.

Candidate evaluation and metric computation. Starting from the N_T-th layer, all constellation symbols are considered as candidates. For each candidate, the squared Euclidean distance (branch metric) is computed relative to the transformed received signal and the upper-triangular matrix.

Adaptive thresholding and pruning. Once the branch metrics are computed, a dynamic threshold η_i is calculated at each layer to eliminate the unlikely candidates. The proposed threshold formulation incorporates three components: the mismatch between the Babai point and the transformed received symbol, a normalized interference term derived from the structure of the upper triangular matrix, and an SNR-dependent denominator that adaptively tightens the threshold magnitude under high SNR conditions. Complex LLL reduction is applied locally within the threshold computation to improve pruning reliability. Symbol candidates with branch metrics exceeding η_i are pruned. This process is repeated from layer N_T down to layer 1.

Path selection. After pruning is applied to layer N_T down to layer 1, each surviving path corresponds to a complete symbol vector. Among these, the candidate with the lowest branch metric in layer 1 is selected as the final estimate, and the corresponding symbol vector is reconstructed by tracking the selected path.

The pseudocode presented as Algorithm 1 describes the detection procedure with adaptive pruning applied at each layer. Branch metrics are computed for candidate symbols and compared against a dynamic threshold which has been derived from the Babai point, the upper triangular matrix, and the SNR. This selective pruning reduces computational complexity by eliminating unlikely candidates early in the search.

The proposed algorithm dynamically adjusts the number of candidates at each detection layer based on a symbol-wise reliability metric and an interference-sensitive threshold. The threshold formulation incorporates SNR normalization, enabling the algorithm to prune the candidate paths more aggressively when the symbol estimate is deemed highly reliable and to retain more paths when uncertainty is greater. This adaptive mechanism achieves a favorable balance between detection performance and computational complexity.

3.3. Complexity Analysis

Computational complexity in MIMO detection algorithms can be evaluated using different metrics, such as execution time (latency) or analytical expressions like Big-$\mathcal{O}$ notation. In this work, complexity is quantified in terms of the number of SED computations, which directly reflects the effort required in evaluating candidates during detection. This metric provides a practical measure of computational load and allows meaningful comparison between different detection schemes from a simulation-based perspective.

In this work, complexity is reported as the number of SED evaluations. Each SED evaluation represents a metric computation triggered by the expansion of the candidate and reflects the actual search workload. Because it is independent of the computing platform and memory configuration, the SED count provides a consistent indicator of computational demand and is therefore more reliable than runtime measurements, which can vary with simulation environment and system resources.

In ML and near-ML detections, the primary contributor to computational complexity is the repeated evaluation of SEDs derived from the ILS equation. The complexity of the proposed method is expressed in terms of total SED calculations and compared with conventional QRD-M detection, forming a basis for the subsequent performance–complexity trade-off analysis.

4. Results and Discussion

This section presents the symbol error rate (SER) performance and computational complexity of the proposed adaptive QRD-M detection method. Simulation results are provided to evaluate the method under various MIMO configurations and modulation schemes. Performance benchmarking includes sphere decoding (SD) for SER only, while both performance and complexity are compared against conventional QRD-M to isolate the effect of the proposed adaptive threshold.

Particular attention is given to the impact of the adaptive threshold on pruning behavior and its influence on performance across different SNR values. Table 1 lists the core simulation parameters used in the performance evaluation and complexity analysis.

Figure 1 illustrates SER performance of conventional QRD-M detection compared to SD in a 4 × 4 MIMO system with 8-PAM modulation. The SD curve, adapted from [13], represents near-ML performance. With increasing M, QRD-M approaches SD. At maximum list size $M = 8$, QRD-M is equivalent to SD (near-ML) reference. This demonstrates that a sufficiently large M enables QRD-M to approximate ML accuracy. In contrast, lower values (e.g. $M = 4$) show noticeable performance degradation at higher SNRs. This comparison validates the use of SD as a reference to assess the effectiveness of suboptimal detection algorithms.

Figure 2 presents SER performance of the proposed BIA-QRD-M method compared to conventional QRD-M with fixed M values of 8, 12, and 16 in a 4 × 4 MIMO system using

Algorithm 1 Pseudocode for the proposed BIA-QRD-M

Input: $H, y, PAM_{table}, M_{init}, M_{min}, \gamma, \delta, \alpha$
Output: estimated symbol vector $\hat{x}$

$[Q, R] = \mathrm{QR}(H)$ ▷ QR decomposition for detection
$\hat{y} = Q^H y$ ▷ Transformed received vector for metric computation
$[\tilde{Q}, \tilde{R}, T] = \mathrm{LLL}(H)$ ▷ Complex LLL for threshold computation
$\tilde{y} = \tilde{Q}^H y$ ▷ Transformed received vector for threshold computation
Step 1: Root layer (layer N_T)
for each symbol x in the `PAM_table` **do**
  Initialize path with symbol x at layer N_T
  $d_i = |\hat{y}_{N_T} - R_{N_T,N_T} x|^2$ ▷ Compute branch metric
  $\hat{b} = \mathrm{round}\left(\frac{\tilde{y}_{N_T}}{\tilde{R}_{N_T,N_T}}\right)$ ▷ Babai estimate
  $interference = 0$
  $\eta_i = \frac{\gamma\,|\hat{b} - \tilde{y}_{N_T}| + \delta\, interference}{1 + \alpha\, SNR_{linear}}$ ▷ Threshold
  if $d_i \leqslant \eta_i$ **then**
    Add candidate to `surviving_paths`
  end if
  if $\mathrm{length}(\texttt{surviving\_paths}) < M_{min}$ **then**
    $M = M_{min}$
  else $M = \mathrm{length}(\texttt{surviving\_paths})$
  end if
  Store the top M surviving candidates for extension at the next layer
end for
Step 2: Remaining layers (from $N_T - 1$ to 1)
for layer = $N_T - 1$ down to 1 **do**
  for each extended candidate ($x \in$ `PAM_table`) from layer $+1$ **do**
    $\textit{contribution\_from\_lower\_layers} = \sum_{j=layer+1}^{N_T} R_{layer,j} \hat{x}_j$
    $d_i = |\, \hat{y}_{\text{layer}} - R_{\text{layer,layer}} x_{\text{layer}} - \textit{contribution\_from\_lower\_layers}\,|^2$
    $\textit{babai\_term} = \sum_{j=\text{layer}+1}^{N_T} \tilde{R}_{\text{layer},j}\, \hat{x}_j$
    $\hat{b} = \mathrm{round}\left(\frac{\tilde{y}_{\text{layer}} - \textit{babai\_term}}{\tilde{R}_{\text{layer, layer}}}\right)$
    $\textit{interference} = \frac{\sum_{j=\text{layer}+1}^{N_T} \left|\tilde{R}_{\text{layer},j}\right|}{\left|\tilde{R}_{\text{layer, layer}}\right|}$
    $\eta_i = \frac{\gamma\,|\hat{b} - \tilde{y}_{\text{layer}}| + \delta\, interference}{1 + \alpha\, SNR_{linear}}$ ▷ Threshold
    if $d_i \leqslant \eta_i$ **then**
      Add candidate to `surviving_paths`
      if $\mathrm{length}(\texttt{surviving\_paths}) < M_{min}$ **then**
        $M = M_{min}$
      else
        $M = \mathrm{length}(\texttt{surviving\_paths})$
      end if
    end if
  end for
  Store the top M surviving candidates for extension at the next layer
end for
Step 3: Final decision
At layer 1, select the candidate with the minimum branch metric
Reconstruct the full symbol vector $\hat{x}$ from the selected path
Return $\hat{x}$
End

Tab. 1. Simulation parameters.

Parameter	Value
No. of subcarries (FFT size)	64
OFDM symbols per frame	14, used only for Monte Carlo averaging, not per-symbol detection
Channel model	Rayleigh flat-fading (1 tap i.i.d.) and frequency-selective (5 taps, Rayleigh fading with exponential power-delay profile)
Number of channel taps	5
Modulation scheme	16-QAM, 64-QAM
MIMO system	4 × 4, 8 × 8
Detection methods	Conventional QRD-M, proposed BIA-QRD-M

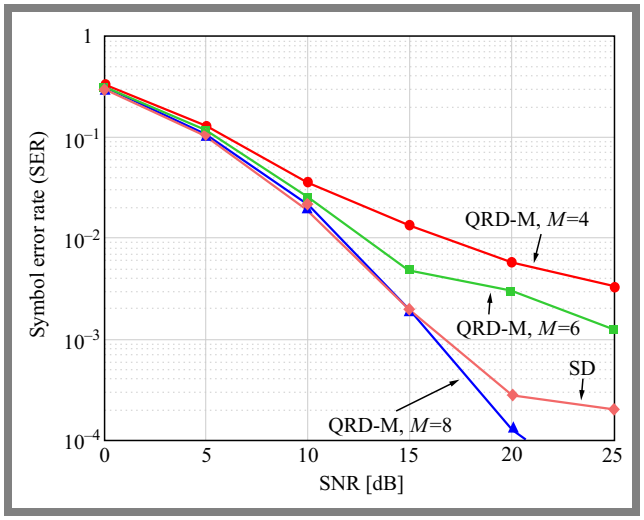

Fig. 1. Comparison of QRD-M detection with M = 4, 6, and 8, and SD in a 4 × 4 MIMO system using 8-PAM modulation.

16-QAM modulation. A larger M in QRD-M detection generally improves accuracy by retaining more candidate paths. In this 16-QAM scenario, the proposed method achieves SER performance that closely matches that of conventional QRD-M with $M = 12$. This demonstrates that the proposed approach achieves near-optimal detection performance while significantly reducing computational complexity, particularly in the medium-to-high SNR regime.

Figure 3 illustrates SER versus SNR for various MIMO configurations and modulation schemes of the proposed BIA-QRD-M system. As shown in Fig. 3, the 4 × 4 MIMO system employing 64-QAM achieves superior SER performance compared to the 4 × 4 MIMO system with 16-QAM, highlighting the advantage of higher-order modulation in terms of detection accuracy.

However, this improvement comes at the expense of increased computational complexity (see subsequent results). Furthermore, comparing 16-QAM, the 8 × 8 MIMO demonstrates enhanced SER performance due to the increased spatial diversity, although with a significantly higher computational cost.

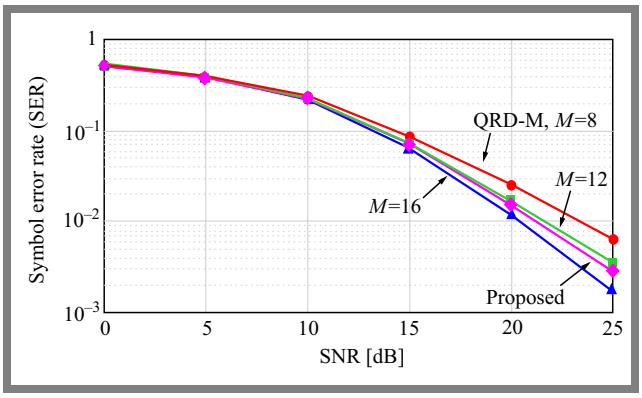

Fig. 2. SER of conventional QRD-M detection with M = 8, 12, and 16, versus the proposed method in a 4 × 4 MIMO system using 16-QAM modulation.

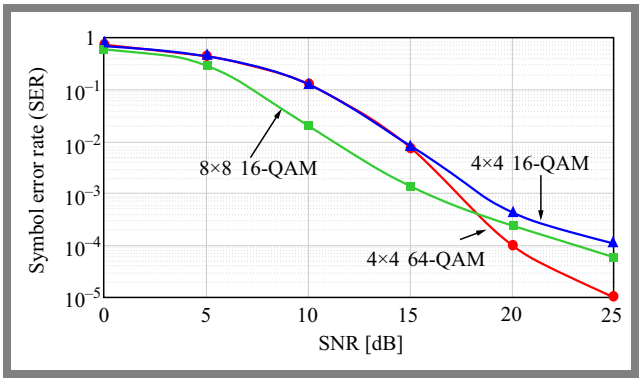

Fig. 3. SER performance of the proposed BIA-QRD-M detection method for 4 × 4 and 8 × 8 MIMO-OFDM systems using 16-QAM and 64-QAM modulation schemes over a flat-fading channel.

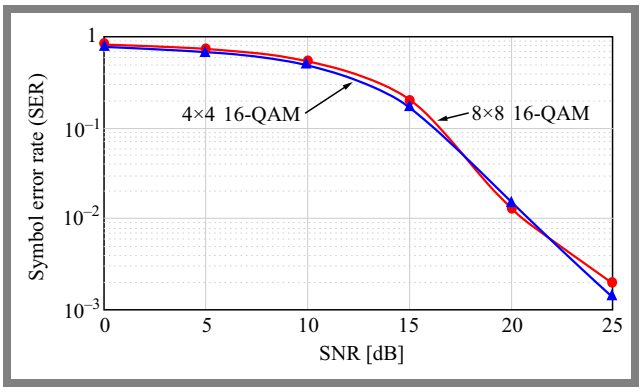

Fig. 4. SER performance of the proposed BIA-QRD-M detection method for 4 × 4 and 8 × 8 MIMO-OFDM systems using 16-QAM modulation over a frequency-selective fading channel.

Figure 4 shows SER performance of the proposed BIA-QAD-M detection method for 4 × 4 and 8 × 8 MIMO-OFDM systems over frequency-selective Rayleigh fading channels. In both configurations, SER decreases consistently with increasing SNR, showing that the proposed method maintains reliable detection accuracy across a wide SNR range.

Following the analysis of the symbol error rate performance, the subsequent focus is on computational complexity. The

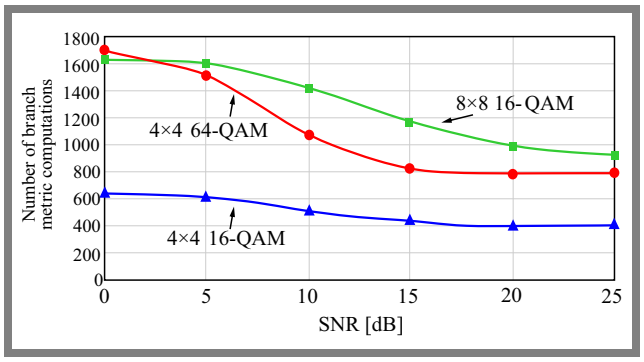

Fig. 5. Average number of branch-metric computations per detection versus SNR for the proposed BIA-QRD-M algorithm in 4 × 4 and 8 × 8 MIMO-OFDM systems using 16-QAM and 64-QAM modulation schemes over a flat-fading channel.

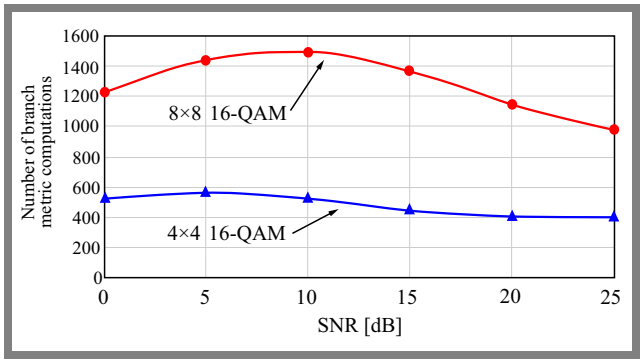

Fig. 6. Average number of branch-metric computations per detection versus SNR for the proposed BIA-QRD-M algorithm in a 4 × 4 and 8 × 8 MIMO-OFDM systems using 16-QAM modulation over a frequency-selective fading channel.

computational cost of the proposed BIA-QRD-M detection algorithm is evaluated under various modulation schemes and MIMO configurations, measured in terms of the average number of branch-metric computations.

The average number of branch metric computations as a function of SNR for the proposed method in 4 × 4 and 8 × 8 MIMO systems using 16-QAM and 64-QAM modulation schemes is presented in Fig. 5. As expected, the computational complexity increases with modulation order. The results confirm that both higher-order modulation and increased antenna count lead to higher computational demands, which is consistent with theoretical expectations.

Figure 6 presents a complexity analysis of the proposed BIA-QRD-M detection method for 4 × 4 and 8 × 8 MIMO-OFDM systems under frequency-selective fading conditions. The results show that the proposed method achieves a substantial reduction in the average number of branch metric computations compared with the conventional QRD-M approach, particularly in the high-SNR region. This reduction is more pronounced for the frequency selective channel due to the enhanced pruning effectiveness at higher SNR, confirming the method's ability to maintain detection accuracy while significantly lowering computational requirements.

A comparison of the average number of branch-metric computations for the proposed method with conventional QRD-M detection using fixed m values of 8, 12, and 16 in a 4 × 4 MIMO system with 16-QAM modulation is illustrated in Fig. 7. Although conventional QRD-M exhibits constant complexity regardless of SNR, the proposed method demonstrates a clear SNR-dependent reduction in computational cost. At high SNR levels, the proposed method approaches the complexity of QRD-M with $M = 8$, the lowest fixed setting, while at low SNR, it maintains a complexity level below that of QRD-M with $M = 12$. This adaptive behavior enables significant complexity savings while preserving detection performance, making it especially attractive for scenarios with dynamic channel conditions.

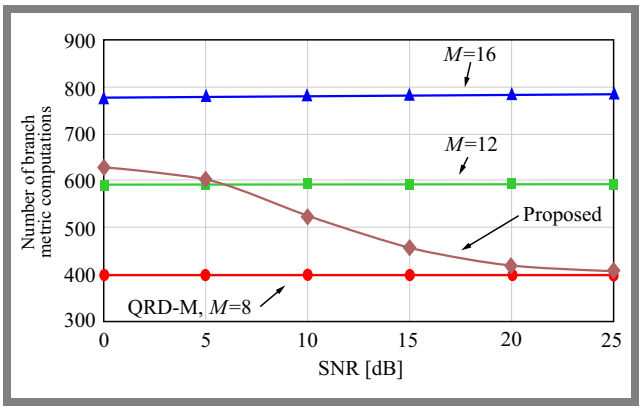

Fig. 7. Comparison of the average number of branch metric computations of conventional QRD-M detection with M = 8, 12, 16, and the proposed method in a 4 × 4 MIMO system with 16-QAM.

To summarize the performance–complexity trade-off, see Tab. 2 presenting a comparative analysis of the proposed and conventional QRD-M methods under various MIMO configurations and modulation schemes at low (10 dB), moderate (15 dB), and high (25 dB) SNR levels. Under 16-QAM modulation, the proposed method consistently outperforms the conventional QRD-M in both 4 × 4 and 8 × 8 MIMO settings. At 10 dB, the 4 × 4 system achieves a lower SER (0.1214 vs. 0.1341) while reducing the branch metric count by more than 20% (424 vs. 541). This efficiency becomes more prominent at 15 dB and 25 dB, where the proposed algorithm cuts complexity by nearly 50% in the 4 × 4 case and by more than 35% in the 8 × 8 case, without compromising SER performance.

For 64-QAM modulation in the 4 × 4 configuration, the trade-off becomes even more evident. At 10 dB, the proposed method significantly reduces SER (0.1250 vs. 0.5697) while requiring only 8.5% of the metric computations (1068 vs. 12 508), indicating substantial gains in both accuracy and efficiency. These benefits are maintained at higher SNR levels. At 15 dB, the SER drops below 10^{-3} with only 818 metric evaluations, i.e. far below the conventional method's complexity threshold.

Across all scenarios shown in Tab. 2, the proposed interference-sensitive pruning strategy enables scalable complexity control while maintaining high detection accuracy. The improvements are particularly pronounced under high-order modulation and larger MIMO sizes, confirming effectiveness in the performance-complexity trade-off for MIMO-OFDM systems.

Tab. 2. Performance–complexity trade-off of conventional and proposed QRD-M detection methods under various MIMO configurations and modulation schemes.

SNR [dB]	MIMO	Modulation	Complexity (QRD-M)	Complexity (proposed)	SER (QRD-M)	SER (proposed)
10	4 × 4	16-QAM	784	518	0.222	0.1315
15	4 × 4	16-QAM	784	425	0.07625	8.398×10^{-3}
25	4 × 4	16-QAM	784	400	2.75×10^{-3}	1.116×10^{-4}
10	4 × 4	64-QAM	12 352	1068	0.5697	0.1250
15	4 × 4	64-QAM	12 352	818	0.3725	0.0079
25	4 × 4	64-QAM	12 352	784	0.0295	0
10	8 × 8	16-QAM	1808	1420	0.1003	0.01965
15	8 × 8	16-QAM	1808	1163	0.0115	0.0013
25	8 × 8	16-QAM	1808	923	8.75×10^{-4}	5.859×10^{-5}

5. Conclusions

This paper presents an adaptive QRD-M detection algorithm enhanced by an interference-aware pruning strategy, designed for scalable and efficient MIMO-OFDM systems. By dynamically adjusting the candidate set based on symbol reliability and inter-layer interference, the proposed method reduces computational complexity while maintaining near-MLD performance. Unlike previous approaches that rely on fixed m configurations, modulation-specific heuristics, or noise-dependent tuning, the proposed scheme adapts to the detection structure itself, ensuring robustness across a wide range of MIMO sizes and modulation formats. The simulation results confirmed that the proposed method improves both detection accuracy and computational efficiency, especially in large-scale and high-order MIMO settings.

These results affirm the practicality for modern wireless systems that require high spectral efficiency under constrained computational resources. This paper reports evaluations for 4 × 4 and 8 × 8 configurations, reflecting the scope commonly adopted in non-linear MIMO detection studies.

References

[1] K. Miura, "An Introduction to Maximum Likelihood Estimation and Information Geometry", *Interdisciplinary Information Sciences*, vol. 17, pp. 155–174, 2011 (https://doi.org/10.4036/iis.2011.155).

[2] Q. Zhou and X. Ma, "Element-based Lattice Reduction Algorithms for Large MIMO Detection", *IEEE Journal on Selected Areas in Communications*, vol. 31, pp. 274–286, 2013 (https://doi.org/10.1109/JSAC.2013.130215).

[3] M.O. Damen, H. El Gamal, and G. Caire, "On Maximum-likelihood Detection and the Search for the Closest Lattice Point", *IEEE Transactions on Information Theory*, vol. 49, pp. 2389–2402, 2003 (https://doi.org/10.1109/TIT.2003.817444).

[4] W.H. Chin, "QRD Based Tree Search Data Detection for MIMO Communication Systems", *2005 IEEE 61st Vehicular Technology Conference*, Stockholm, Sweden, 2005 (https://doi.org/10.1109/VETECS.2005.1543595).

[5] M. Mohaisen and K. Chang, "Upper-lower Bounded-complexity QRD-M for Spatial Multiplexing MIMO-OFDM Systems", *Wireless Personal Communications*, vol. 61, pp. 129–141, 2011 (https://doi.org/10.1007/s11277-010-0014-8).

[6] U. Ummatov and K. Lee, "Adaptive Threshold-aided K-best Sphere Decoding for Large MIMO Systems", *Applied Sciences*, vol. 9, art. no. 4624, 2019 (https://doi.org/10.3390/app9214624).

[7] J.-H. Ro, J.-K. Kim, Y.-H. You, and H.-K. Song, "Low-complexity QRD-M with Path Eliminations in MIMO-OFDM Systems", *Applied Sciences*, vol. 7, art. no. 1206, 2017 (https://doi.org/10.3390/app7121206).

[8] S.-J. Choi *et al.*, "Novel MIMO Detection with Improved Complexity for Near-ML Detection in MIMO-OFDM Systems", *IEEE Access*, vol. 7, pp. 60389–60398, 2019 (https://doi.org/10.1109/ACCESS.2019.2914707).

[9] B.S. Kim, S.D. Kim, D. Na, and K. Choi, "A Very Low Complexity QRD-M MIMO Detection Based on Adaptive Search Area", *Electronics*, vol. 9, art. no. 756, 2020 (https://doi.org/10.3390/electronics9050756).

[10] H. Liu *et al.*, "A Novel Iterative Detection Method Based on a Lattice Reduction-aided Algorithm for MIMO OFDM Systems", *Scientific Reports*, vol. 14, art. no. 2779, 2024 (https://doi.org/10.1038/s41598-024-52602-6).

[11] X. Zhou *et al.*, "Model-driven Deep Learning-based MIMO-OFDM Detector: Design, Simulation, and Experimental Results", *IEEE Transactions on Communications*, vol. 70, pp. 5193–5207, 2022 (https://doi.org/10.1109/TCOMM.2022.3186404).

[12] T.-D. Chiueh, P.-Y. Tsai, and I.-W. Lai, *Baseband Receiver Design for Wireless MIMO-OFDM Communications*, Wiley, 346 p., 2012 (https://doi.org/10.1002/9781118188194).

[13] A. Ghasemmehdi and E. Agrell, "Faster Recursions in Sphere Decoding", *IEEE Transactions on Information Theory*, vol. 57, pp. 3530–3536, 2011 (https://doi.org/10.1109/TIT.2011.2143830).

Mar Mar Lwin, Student
School of Electrical and Electronic Engineering
https://orcid.org/0009-0009-3920-5197
E-mail: mar.mar.lwin@student.usm.my
Universiti Sains Malaysia, Nibong Tebal, Malaysia
https://www.eng.usm.my

Mohd Fadzli Mohd Salleh, Ph.D., Assoc. Professor
School of Electrical and Electronic Engineering
https://orcid.org/0000-0002-1801-6049
E-mail: fadzlisalleh@usm.my
Universiti Sains Malaysia, Nibong Tebal, Malaysia
https://www.eng.usm.my

Half-duplex Two-way Relaying for Wireless Sensor Networks with Adaptive Coding Rate: A Performance Optimization Framework

The-Anh Ngo[1], Viet-Thanh Le[2], Thien P. Nguyen[2], and Duy-Hung Ha[2]

[1] *University of Transport and Communications, Ho Chi Minh City, Vietnam,*
[2] *Ton Duc Thang University, Ho Chi Minh City, Vietnam*

https://doi.org/10.26636/jtit.2025.4.2314

Abstract — In this paper, a novel framework to enhance the reliability of wireless sensor networks (WSNs) by addressing the high probability of outage (OP) resulting from limited energy resources and unreliable channels. The framework integrates three techniques: half-duplex two-way relaying (HD-TWR), digital network coding (DNC), and rateless codes. Although these techniques have been extensively studied in isolation, a comprehensive analysis of their joint performance is provided as the main contribution. The proposed scheme leverages the energy efficiency of HD-TWR, the transmission reduction capability of DNC, and the retransmission-free resilience of rateless codes. Simulation results show that the integrated framework significantly reduces OP, offering a robust and practical solution to enhance the reliability enhancement. Furthermore, the impact of optimal relay node placement is investigated through parameter adjustments in the simulation stage to maximize performance gains.

Keywords — digital network coding, half-duplex two-way relaying, outage probability, rateless codes, relay placement, wireless sensor networks

1. Introduction

In the past decade, wireless sensor networks (WSNs) have become a foundational technology for real-time monitoring and data acquisition in a wide range of applications, from environmental surveillance to industrial automation. Comprised of numerous low-power sensor nodes, WSNs are crucial enablers of the Internet of Things (IoT) [1]–[3]. However, the performance of these networks is limited by their resource-constrained environments, energy resources, inherent unreliability, and security vulnerabilities of wireless channels. These limitations collectively lead to critical performance issues, particularly high outage probability (OP) and loss of secrecy, which compromise network reliability and data confidentiality. Therefore, developing robust, energy efficient, and secure communication protocols is a paramount challenge in WSNs [4]–[6].

To address the mentioned limitations, various advanced communication strategies have been investigated. First, node deployment techniques to enhance network coverage and data security have been studied in [3], [5], [7], [8]. Next, cooperative relaying has emerged as a key solution for improving network performance, such as throughput and outage probability as well as network lifetime, end-to-end delay, and secrecy [9]–[12]. While full-duplex relaying offers significant gains in spectral efficiency, its practical implementation is often hindered by the complex and power-intensive problem of mitigating residual self-interference cancelation.

On the contrary, half-duplex (HD) operation, which avoids simultaneous transmission and reception, inherently bypasses this issue [13]–[16]. HD relaying is simpler and more energy-efficient than their full-duplex counterparts as they only transmit or receive at any given time, making them a suitable choice for WSNs [16]. Moreover, a combination of HD and two-way relaying (TWR) can be useful to improve network performance [17], [18].

On a more fundamental level, digital network coding (DNC) is a powerful principle to enhance network efficiency. Rather than simply forwarding packets, DNC allows an intermediate node to combine data from multiple incoming streams before transmission (e.g., using an XOR operation) [19]. This technique significantly reduces the number of transmissions required to exchange information, thereby improving both network throughput and spectral efficiency.

To further bolster network reliability against channel impairments, rateless coding offers a decent solution. Unlike conventional fixed-rate codes that require a reliable channel to be effective, rateless codes (RC) allow a source to generate an infinite stream of encoded symbols, ensuring that a receiver can successfully decode the original message as soon as a sufficient number of symbols are collected. This property makes RC highly suitable for dynamic and unpredictable as well as unsecured wireless environments [20]–[24].

Although significant progress has been achieved, the aforementioned studies have been carried out independently. Building on these advancements, this work proposes a novel communication framework that integrates HD-TWR, DNC, and RC strategies. The core objective of this research is to signifi-

cantly minimize OP by also considering the optimal placement of the relay node. We develop a comprehensive analytical framework to model system outage performance for various network scenarios.

The key contributions of this work are the integration of rateless coding with a practical DNC-based HD-TWR scheme and a simulation-based demonstration of how the position of the relay affects system performance.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the system model. Section 4 provides the outage performance analysis. Section 5 presents the simulation results and discussions. Finally, Section 6 concludes the article and outlines potential future work.

2. Related Works

Research on enhancing the performance of wireless communication networks, particularly wireless sensor networks (WSNs), has evolved along several key directions, most notably in the areas of relaying strategies, advanced coding schemes, and network optimization. This section reviews the most pertinent literature related to wireless communications and WSNs, emphasizing the contributions and limitations of existing studies that motivate the integrated approach advanced in this paper.

2.1. Cooperative Relaying and HD-TWR

The fundamental principles of half-duplex relaying have been studied in prior work [15], [16], [25]–[27], which demonstrate that separating transmission and reception into orthogonal slots mitigates self-interference and simplifies transceiver design. In the simplest HD model, a relay node (RN) facilitates communication between two source nodes, requiring three time slots for the corresponding transmission phases, since the RN cannot transmit and receive simultaneously. These studies suggest that HD relaying can improve energy efficiency, transmission rate, probability of outage, and reliability in resource-constrained wireless communication systems. However, they remain largely limited to basic performance evaluations without addressing broader efficiency considerations.

Building on this foundation, subsequent research on half-duplex two-way relaying (HD-TWR) [17], [18], [28] investigates bidirectional communication, thereby reducing latency and improving spectral efficiency compared to conventional one-way relaying. Although these studies report notable performance gains, they also reveal persistent limitations. The half-duplex constraint inherently reduces throughput compared to full-duplex systems, and the effectiveness of HD-TWR is further challenged by channel estimation errors, synchronization difficulties, and relay processing overhead.

Furthermore, none of the existing HD-TWR works mentioned above has explored the integration of advanced coding techniques such as rateless codes and digital network coding, which holds significant promise for enhancing adaptability, reliability, and overall network performance. This research gap indicates that current HD and HD-TWR frameworks are insufficient to provide scalable, secure, and energy-efficient solutions for WSNs and next-generation wireless networks, thus motivating integrated approaches that jointly exploit relaying, coding, and network optimization to achieve practical improvements in efficiency, robustness, scalability, and secrecy performance.

2.2. Digital Network Coding and Rateless Codes

Digital network coding, typically implemented through XOR operations at the relay, reduces the number of transmission phases by combining packets from different sources, thus improving throughput, spectral efficiency, and latency in wireless communication systems [18], [19]. Despite these benefits, DNC remains sensitive to synchronization errors, imperfect channel estimation, and error propagation, which limit its robustness in practical scenarios.

Rateless codes extend the principle to generate an unlimited stream of coded packets, allowing receivers to decode once a sufficient number of symbols has been collected [29]. In general, RCs encompass a broad class of coding schemes such as Luby transform (LT), raptor, and random linear network coding (RLNC). In RLNC, each encoded packet is a random linear combination of source packets over a finite field.

In this paper, we consider a generic RC model without specifying the exact encoding structure, focusing on OP performance rather than decoding complexity. This general formulation allows the proposed framework to capture the performance behavior of various RC schemes without being restricted to a specific encoding algorithm. This rateless property not only enhances adaptability, reliability, and outage performance in time-varying channels but also provides inherent physical layer security (PLS), since eavesdroppers who do not accumulate enough packets cannot reconstruct the original data.

Recent studies confirm that CR can improve secrecy capacity, increase secrecy throughput, and reduce the probability of secrecy outages by exploiting channel asymmetry [18], [20], [22], [24], [30]. However, RC also suffers from decoding complexity, signaling overhead, and delay, particularly in large-scale networks. Although the integration of RC and DNC in full-duplex two-way relaying (FD-TWR) has been investigated [18], OP analysis has not been addressed. This gap provides the main motivation for the present work.

2.3. Relay Node Placement

Node deployment and relay selection are fundamental design aspects in cooperative wireless networks, as they directly influence coverage, reliability, and spectral efficiency [3], [5], [7], [8], [31]. However, none of the existing studies have evaluated network performance in terms of OP for HD-TWR systems that employ integrated RC and DNC. Given that OP is a critical metric for assessing link reliability under realistic channel conditions, its absence in previous research represents a significant gap. Therefore, it is essential

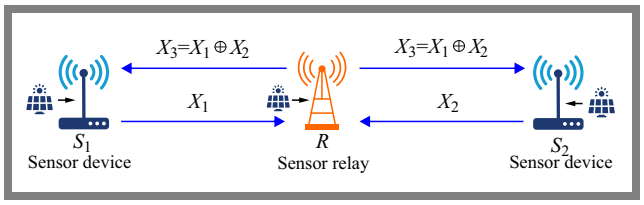

Fig. 1. The HD-TWR model using DNC and RC.

to conduct a comprehensive study on OP improvement of OP in HD-TWR systems with RC-DNC, providing deeper insight into system performance and reliability.

In summary, although HD-TWR has been widely studied, existing works have not incorporated advanced coding techniques such as RC and DNC, which could jointly improve adaptability, efficiency, and secrecy performance. On the contrary, research on RC and DNC has primarily addressed throughput, spectral efficiency, or physical layer security, but has not considered their integration into HD-TWR frameworks. Additionally, these studies typically overlook critical system-level aspects such as relay node deployment and placement optimization, which have a direct impact on coverage, reliability and OP. This gap indicates that current approaches remain insufficient to achieve robust and scalable performance in realistic network scenarios.

Motivated by these limitations, this article investigates the joint application of RC and DNC in HD-TWR systems, with a particular focus on evaluating and improving OP under different operating conditions. Additionally, we analyze the effect of relay node placement through simulation-based parameter adjustment, providing insights into optimal deployment strategies for improved system performance.

The relay node is assumed to be located along the line between S_1 and S_2. Let x_R denote its normalized position, where $x_R = 0$ and $x_R = 1$ correspond to the locations of S_1 and S_2, respectively. The effect of the placement of the relay on the outage probability is analyzed by varying x_R in the range $\{0,1\}$.

3. System Model

The system model using the HD-TWR relaying strategy combined with DNC-RC techniques (denoted as SM3P) is shown in Fig. 1. Two nodes, S_1 and S_2, exchange data via a relay R. Using RC, S_1 and S_2 split the original message into equal-size packets and XOR one or multiple selected packets to produce encoded packets.

Figure 1 depicts the process of exchanging encoded packets between S_1 and S_2, where $x_1(x_2)$ represents an encoded packet of $S_1(S_2)$ that needs to be sent to $S_2(S_1)$. Specifically, in the first phase, S_1 transmits x_1 to R, and in the subsequent phase, S_2 transmits x_2 to R. If R successfully decodes both x_1 and x_2, it performs the XOR operation as follows: $x_3 = x_1 \oplus x_2$. Subsequently, R transmits x_3 to both S_1 and S_2. Upon successfully decoding x_3, $S_1(S_2)$ can retrieve the desired $x_2(x_1)$ using the operation: $x_1 \oplus x_3 = x_2$ $(x_2 \oplus x_3 = x_1)$.

To successfully recover each other's original information, S_1 and S_2 are assumed to receive at least H packets without errors. Moreover, due to latency constraints, S_1 and S_2 can attempt to exchange their information packets x_1 and x_2 through at most Q transmission rounds, where $Q \geqslant H$. The transmission is considered successful if at least H out of Q encoded packets are correctly received and decoded. Indeed, upon the completion of data transmission, if $S_1(S_2)$ has not received the required number of packets, it will fail to recover the original information of $S_2(S_1)$, leading to an outage. It is also assumed that devices S_1, S_2, and R are equipped with a single antenna and that all transmission channels experience Rayleigh fading.

Additionally, this paper considers block fading channels, where the channel between two nodes remains constant within a phase but changes independently in subsequent transmission phases.

Considering data transmission in phase 1, the channel capacity between S_1 and R is given by:

$$C_{S_1 \to R} = \frac{1}{3} \log_2 \left(1 + \frac{P_1 \gamma_{S_1 \to R}}{\sigma_0^2} \right), \tag{1}$$

where, $\frac{1}{3}$ indicates that the transmission of each encoded packet occurs over three orthogonal time slots. The transmit power of S_1 is denoted as P_1, while $\gamma_{S_1 \to R}$ represents the channel gain between the nodes in phase 1. Additionally, σ_0^2 denotes the power of the additive white Gaussian noise (AWGN) at R (as well as at other receiving devices).

If $C_{S_1 \to R} \geqslant C_{th}$, we assume that R can successfully decode x_1, where C_{th} is a predefined threshold. Conversely, if $C_{S_1 \to R} < C_{th}$, x_1 cannot be decoded at R. Similarly, the channel capacity between S_2 and R is given by:

$$C_{S_2 \to R} = \frac{1}{3} \log_2 \left(1 + \frac{P_2 \gamma_{S_2 \to R}}{\sigma_0^2} \right), \tag{2}$$

where P_2 is the transmit power of S_2, and $\gamma_{S_2 \to R}$ represents the channel gain between the nodes in phase 2.

Similar to x_1, if $C_{S_2 \to R} \geqslant C_{th}$, the packet x_2 is successfully decoded at node R. Otherwise, node R fails to decode x_2.

Consequently, there exist four distinct cases regarding the decoding capability of node R for x_1 and x_2:

- **Case 1:** Neither x_1 nor x_2 is successfully decoded.
  In this scenario, $C_{S_1 \to R} < C_{th}$ and $C_{S_2 \to R} < C_{th}$. As a result, node R cannot transmit any encoded packet in phase 3 since both x_1 and x_2 are erroneously decoded (i.e., sources S_1 and S_2 do not receive any encoded packet).
- **Case 2:** x_1 is successfully decoded, while x_2 is not.
  In this case, $C_{S_1 \to R} \geqslant C_{th}$ and $C_{S_2 \to R} < C_{th}$. Consequently, R will transmit x_1 to S_2 in phase 3, and the achievable channel capacity is:

$$C_{R \to S_2} = \frac{1}{3} \log_2 \left(1 + \frac{P_3 \gamma_{R \to S_2}}{\sigma_0^2} \right), \tag{3}$$

  where P_3 is the transmit power of R, and $\gamma_{R \to S_2}$ denotes the channel gain between R and S_2.
- **Case 3:** x_2 is successfully decoded, while x_1 is not.
  In this scenario, $C_{S_1 \to R} < C_{th}$ and $C_{S_2 \to R} \geqslant C_{th}$. Thus,

R will forward x_2 to S_1 in phase 3, and the achievable channel capacity is:

$$C_{R\to S_1} = \frac{1}{3}\log_2\left(1+\frac{P_3\gamma_{R\to S_1}}{\sigma_0^2}\right), \tag{4}$$

where $\gamma_{R\to S_1}$ represents the channel gain between R and S_1.

- **Case 4:** Both x_1 and x_2 are successfully decoded. This case has been described earlier, where $C_{S_1\to R} \geqslant C_{th}$ and $C_{S_2\to R} \geqslant C_{th}$. In this situation, R will transmit x_3 to both S_1 and S_2, and the corresponding channel capacities are given by:

$$\begin{aligned} C_{R\to S_1} &= \frac{1}{3}\log_2\left(1+\frac{P_3\gamma_{R\to S_1}}{\sigma_0^2}\right), \\ C_{R\to S_2} &= \frac{1}{3}\log_2\left(1+\frac{P_3\gamma_{R\to S_2}}{\sigma_0^2}\right). \end{aligned} \tag{5}$$

4. Outage Probability Analysis

In this section, section a mathematical analysis of the outage probability at S_1 and S_2, which represents the probability that $S_1(S_2)$ fails to receive H encoded packets from $S_2(S_1)$. Given that the transmission channels experience Rayleigh fading, the channel gain between the transmitting node A and the receiving node B, where $A, B \in \{S_1, S_2, R\}$, follows the distributions:

$$\begin{aligned} F_{\gamma_{A\to B}}(x) &= 1-\exp(-\lambda_{A,B}x), \\ f_{\gamma_{A\to B}}(x) &= \lambda_{A,B}\exp(-\lambda_{A,B}x), \end{aligned} \tag{6}$$

where $F_{\gamma_{A\to B}}(x)$ and $f_{\gamma_{A\to B}}(x)$ denote the cumulative distribution function (CDF) and probability density function (PDF) of the channel gain $\gamma_{A\to B}$, respectively. Here, $\lambda_{A,B} = (d_{A,B})^\beta$ [22], with $d_{A,B}$ representing the distance between A and B, and β being the path loss exponent, where λ is the exponential parameter. In Rayleigh fading, the instantaneous SNR $\gamma = P|h|^2/N_0$ follows an exponential distribution with mean λ^{-1}. Therefore, λ corresponds to the inverse of the average SNR parameter.

A packet x_2 fails to reach S_1 if at least one of the two links, $S_2 \to R$ or $R \to S_1$, does not meet the required quality, i.e., $C_{S_2\to R} < C_{th}$ or $C_{R\to S_1} < C_{th}$. Consequently, the probability that an encoded packet from S_2 cannot be successfully transmitted to S_1 is expressed as:

$$\begin{aligned} \theta_{S_1} &= \Pr(C_{S_2\to R} \geqslant C_{th} \cup C_{R\to S_1} \geqslant C_{th}) \\ &= 1-\Pr(C_{S_2\to R} > C_{th} \cap C_{R\to S_1} > C_{th}) \\ &= 1-\Pr(C_{S_2\to R} > C_{th})\Pr(C_{R\to S_1} > C_{th}). \end{aligned} \tag{7}$$

Substituting the results from Eqs. (2) and (4) into Eq. (7), we obtain:

$$\begin{aligned} \theta_{S_1} &= 1-\Pr(\gamma_{S_2\to R} > \rho_2)\Pr(\gamma_{R\to S_1} > \rho_3) \\ &= 1-\Big(1-F_{\gamma_{S_2\to R}}(\rho_2)\Big)\Big(1-F_{\gamma_{R\to S_1}}(\rho_3)\Big), \end{aligned} \tag{8}$$

where:

$$\rho_2 = \frac{\left(2^{3C_{th}}-1\right)\sigma_0^2}{P_2}, \quad \rho_3 = \frac{\left(2^{3C_{th}}-1\right)\sigma_0^2}{P_3}. \tag{9}$$

By substituting the CDF functions from Eq. (6) into Eq. (8), we obtain:

$$\theta_{S_1} = 1-\exp\left(-\lambda_{S_2,R}\rho_2\right)\exp\left(-\lambda_{S_1,R}\rho_3\right). \tag{10}$$

Similarly, the probability that an encoded packet from S_1 fails to be successfully transmitted to S_2 is given by:

$$\begin{aligned} \theta_{S_2} &= \Pr\left(C_{S_1\to R} \leqslant C_{th} \cup C_{R\to S_2} \geqslant C_{th}\right) \\ &= 1-\left(1-F_{\gamma_{S_1\to R}}(\rho_1)\right)\left(1-F_{\gamma_{R\to S_1}}(\rho_3)\right) \\ &= 1-\exp\left(-\lambda_{S_1,R}\rho_1\right)\exp\left(-\lambda_{S_1,R}\rho_3\right), \end{aligned} \tag{11}$$

where $\rho_1 = \dfrac{\left(2^{3C_{th}}-1\right)\sigma_0^2}{P_1}$.

After the packet exchange process, let n_1 and n_2 denote the number of packets successfully received at S_1 and S_2, respectively. Considering S_2, the original information from S_1 can be successfully reconstructed if $n_2 \geqslant H$. If $n_2 < H$, then S_2 fails to reconstruct the information, resulting in an outage. Using Eq. (11), the OP at S_2 is derived as:

$$\begin{aligned} OP_{S_2}^{SM3P} &= \sum_{n_2=0}^{H-1} C_Q^{n_2}\left(1-\theta_{S_2}\right)^{n_2}\left(\theta_{S_2}\right)^{Q-n_2} \\ &= \sum_{n_2=0}^{H-1} C_Q^{n_2}\exp\left(-n_2\lambda_{S_1,R}\rho_1 - n_2\lambda_{S_1,R}\rho_3\right) \\ &\quad\times\left[1-\exp\left(-\lambda_{S_1,R}\rho_1-\lambda_{S_1,R}\rho_3\right)\right]^{Q-n_2}, \end{aligned} \tag{12}$$

where $C_Q^{n_2} = \dfrac{Q!}{n_2!\,(Q-n_2)!}$ denotes the binomial coefficient.

Note that any subset of H correctly received packets out of Q total attempts is sufficient for successful decoding.

Similarly, the outage probability at the source S_1 is given by the following exact closed-form expression:

$$\begin{aligned} OP_{S_1}^{SM3P} &= \sum_{n_1=0}^{H-1} C_Q^{n_1}\left(1-\theta_{S_1}\right)^{n_1}\left(\theta_{S_1}\right)^{Q-n_1} \\ &= \sum_{n_1=0}^{H-1} C_Q^{n_1}\exp\left(-n_1\lambda_{S_2,R}\rho_2 - n_1\lambda_{S_1,R}\rho_3\right) \\ &\quad\times\left[1-\exp\left(-\lambda_{S_2,R}\rho_1-\lambda_{S_1,R}\rho_3\right)\right]^{Q-n_1}. \end{aligned} \tag{13}$$

Next, the conventional four-phase HD-TWR model, referred to as SM4P will be analysed. In this model, the first two transmission phases are used to transmit x_1 from S_1 through R to S_2, while the remaining two phases are used to transmit x_2 from S_2 through R to S_1. Due to the four-phase transmission scheme, the channel capacity of the links is determined as follows:

$$\begin{aligned} C_{S_1\to R}^* &= \frac{1}{4}\log_2\left(1+\frac{P_1\gamma_{S_1\to R}}{\sigma_0^2}\right), \\ C_{R\to S_2}^* &= \frac{1}{4}\log_2\left(1+\frac{P_3\gamma_{R\to S_2}}{\sigma_0^2}\right), \\ C_{S_2\to R}^* &= \frac{1}{4}\log_2\left(1+\frac{P_2\gamma_{S_2\to R}}{\sigma_0^2}\right), \\ C_{R\to S_1}^* &= \frac{1}{4}\log_2\left(1+\frac{P_3\gamma_{R\to S_1}}{\sigma_0^2}\right). \end{aligned} \tag{14}$$

Using the same analytical approach as in SM3P, the probabilities that S_1 and S_2 fail to achieve a single encoded packet are given as follows:

$$\begin{aligned}\theta^*_{S_1} &= \Pr\left(C^*_{S_2\to R} \leqslant C_{th} \cup C^*_{R\to S_1} \leqslant C_{th}\right)\\ &= 1-\exp\left(-\lambda_{S_2,R}\theta_2\right)\exp\left(-\lambda_{S_1,R}\theta_3\right),\\ \theta^*_{S_2} &= \Pr\left(C^*_{S_1\to R} \leqslant C_{th} \cup C^*_{R\to S_2} \leqslant C_{th}\right)\\ &= 1-\exp\left(-\lambda_{S_1,R}\theta_2\right)\exp\left(-\lambda_{S_1,R}\theta_3\right),\end{aligned} \tag{15}$$

where:

$$\begin{aligned}\theta_1 &= \frac{(2^{4C_{th}}-1)\sigma_0^2}{P_1},\\ \theta_2 &= \frac{(2^{4C_{th}}-1)\sigma_0^2}{P_2},\\ \theta_3 &= \frac{(2^{4C_{th}}-1)\sigma_0^2}{P_3}.\end{aligned} \tag{16}$$

Subsequently, the OP at S_1 and S_2 in SM4P is respectively given by the following exact closed-form expressions:

$$\begin{aligned}OP_{S_1}^{SM4P} &= \sum_{n_1=0}^{H-1} C_Q^{n_1}\left(1-\theta^*_{S_1}\right)^{n_1}\left(\theta^*_{S_1}\right)^{Q-n_1}\\ &= \sum_{n_1=0}^{H-1} C_Q^{n_1}\exp\left(-n_1\lambda_{S_2,R}\theta_2 - n_1\lambda_{S_1,R}\theta_3\right)\\ &\quad\times\left[1-\exp\left(-\lambda_{S_2,R}\theta_2-\lambda_{S_1,R}\theta_3\right)\right]^{Q-n_1},\end{aligned} \tag{17}$$

$$\begin{aligned}OP_{S_2}^{SM4P} &= \sum_{n_2=0}^{H-1} C_Q^{n_2}\left(1-\theta^*_{S_2}\right)^{n_2}\left(\theta^*_{S_2}\right)^{Q-n_2}\\ &= \sum_{n_2=0}^{H-1} C_Q^{n_2}\exp\left(-n_2\lambda_{S_1,R}\theta_1 - n_2\lambda_{S_1,R}\theta_3\right)\\ &\quad\times\left[1-\exp\left(-\lambda_{S_1,R}\theta_1-\lambda_{S_1,R}\theta_3\right)\right]^{Q-n_2},\end{aligned} \tag{18}$$

Before proceeding to Section 5, we consider the power allocation problem for S_1, S_2, and R, formulated as follows:

$$P_1+P_2+P_3 = P\,. \tag{19}$$

Equation (19) implies that the total transmit power of all nodes is constrained to P, and we need to allocate P_1, P_2, and P_3 to achieve optimal OP performance.

First, considering that S_1 and S_2 are typically identical devices, we assume equal transmit power for both, i.e., $P_1 = P_2 = P_S$. Consequently, Eq. (19) simplifies to $2P_S+P_3 = P$. Therefore, if we set $P_S = \mu P$, the transmit power of R is given by $P_R = P_3 = (1-2\mu)P$, where μ $(0<\mu<0.5)$ is a predetermined coefficient.

5. Simulation Results and Discussions

In this section, we perform Monte Carlo simulations to verify the derived OP formulas, such as (12), (13), (17), and (18).

Consider the axis O_x, where S_1, S_2, and R have coordinates $S_1(0)$, $S_2(1)$, and $R(x_R)$, respectively, with $0<x_R<1$. Given these positions, the distance between S_1 and S_2 is fixed at 1, while R moves between S_1 and S_2. The distances between R and the sources are expressed as $d_{S_1,R} = x_R$ and $d_{S_2,R} = 1-x_R$.

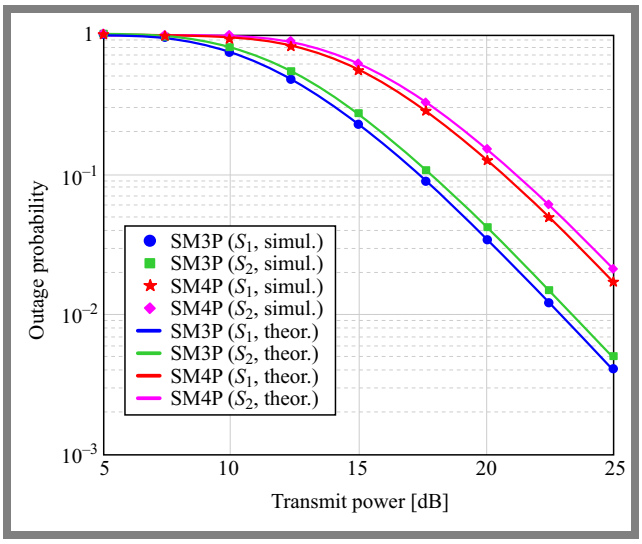

Fig. 2. Outage probability vs. transmit power (in decibels) with $H=4$, $Q=5$, $x_R=0.35$, and $\mu=0.35$.

To focus on investigating the impact of key parameters, we fix the following system parameters: noise power $\sigma_0^2=1$, outage threshold $C_{th}=1$, and path-loss exponent $\beta=3$.

Figure 2 plots the OP of S_1 and S_2 in both SM3P and SM4P models versus P [dB]. In Fig. 2, the system parameters are set as $H=4$, $Q=5$, x_R=0.35, and μ=0.35. With μ=0.35, the transmit powers of the nodes are P_S=0.35P and P_R=0.3P. Figure 2 shows that the OP of both S_1 and S_2 decreases as P increases because higher P leads to higher P_S and P_R. We also observe that the OP of S_1 and S_2 in SM3P is lower than in SM4P since SM3P uses only three transmission phases.

Furthermore, in both SM3P and SM4P, the OP of S_1 is lower than that of S_2. This is because the transmit power of R is smaller than that of the source nodes ($P_S > P_R$). Additionally, R is farther from S_2, so the packet transmission from R to S_2 in phase 3 is less reliable than from R to S_1.

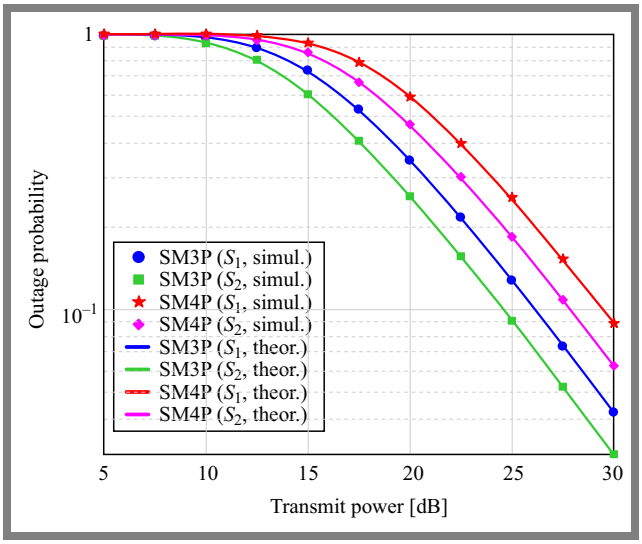

Fig. 3. Outage probability vs. transmit power (in decibels) with $H=5$, $Q=5$, $x_R=0.6$, and $\mu=0.4$

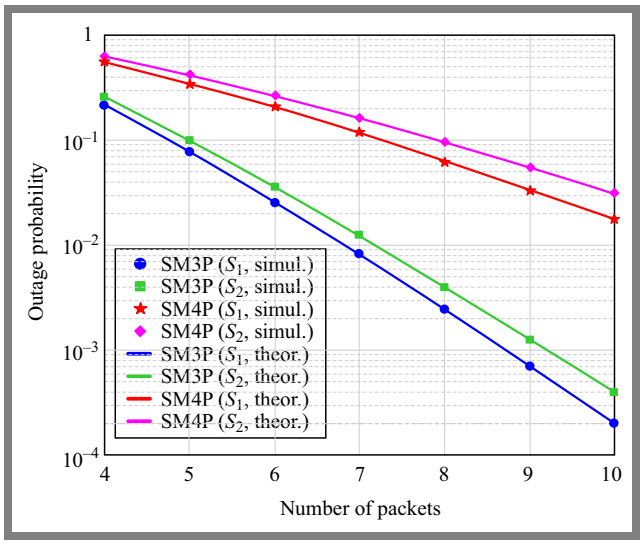

Fig. 4. Outage probability vs. number of packets with $P = 10$ dB, $H = 3$, $x_R = 0.55$, and $\mu = 0.3$.

The results in Fig. 2 also demonstrate excellent agreement between simulation and theory, validating the accuracy of the derived Eqs. (12), (13), (17), and (18) presented in Section 4.

Figure 3 illustrates the OP of S_1 and S_2 in SM3P and SM4P versus P [dB] with $H = 5$, $Q = 5$, $x_R = 0.6$, and $\mu = 0.4$. Similarly, SM3P achieves lower OP at both sources compared to SM4P. However, the OP at S_2 in both SM3P and SM4P is lower than at S_1. As explained for Fig. 2, this is because $P_S = 0.4P > P_R = 0.2P$, and R is closer to S_2 than to S_1, resulting in lower OP at S_2.

Figure 4 plots the OP at the sources as a function of the maximum number of retransmissions Q for $P = 10$ dB, $H = 3$, $x_R = 0.55$, and $\mu = 0.3$. As expected, the OP of all sources decreases as Q increases. Figure 4 also shows that the OP of SM3P and SM4P decreases faster with increasing Q. Moreover, the OP in SM3P decreases more rapidly than in SM4P. Thus, to enhance reliability (i.e., reduce OP) in SM3P and SM4P, we can increase Q. For example, in SM3P,

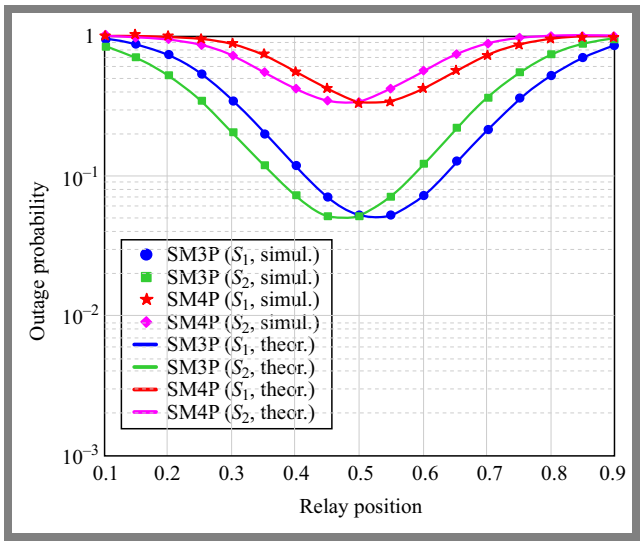

Fig. 5. Outage probability vs. relay position with $P = 10$ dB, $H = 4$, $Q = 7$, and $\mu = 0.3$.

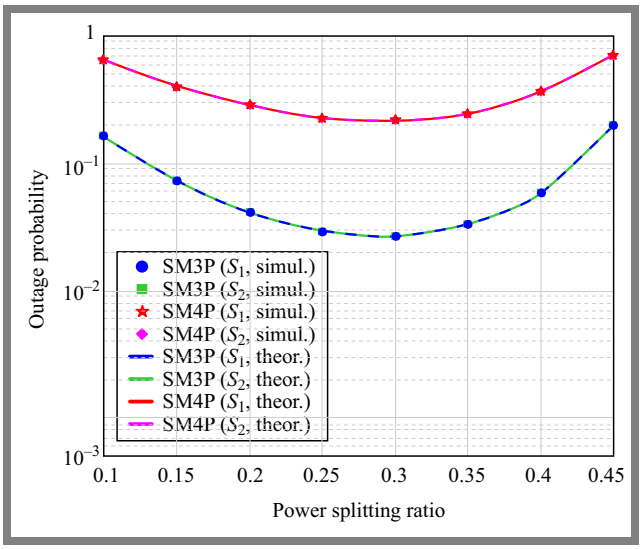

Fig. 6. Outage probability vs. power spliting ratio with $P = 10$ dB, $H = 3$, $Q = 6$, and $x_R = 0.5$.

$Q \geqslant 8$ is required to achieve OP < 0.001 at both sources. However, increasing Q also raises the network delay and energy consumption.

Figure 5 analyzes the impact of relay position x_R on system performance with $P = 10$ dB, $H = 4$, $Q = 7$, and $\mu = 0.3$. The results show that x_R significantly affects the OP at S_1 and S_2 in both SM3P and SM4P. Specifically, when $x_R < 0.5$, the OP at S_2 is lower than at S_1. Conversely, for $x_R > 0.5$, $OP_{S_1}^{SM3P} < OP_{S_2}^{SM3P}$ and $OP_{S_1}^{SM4P} < OP_{S_2}^{SM4P}$. At $x_R = 0.5$ (i.e., R is equidistant to both sources), the OP is balanced and minimized. Thus, placing R midway optimizes outage performance.

Figure 6 investigates the effect of μ on the OP in SM3P and SM4P with $P = 10$ dB, $H = 3$, $Q = 6$, and $x_R = 0.5$. Here, R is centered to ensure symmetric performance. The results reveal that μ critically influences the OP, and an optimal $\mu = \frac{1}{3}$ exists where $P_S = P_R = \frac{P}{3}$, minimizing the OP. Due to the complexity of the OP expressions, a rigorous proof of this optimum will be addressed in future work.

6. Conclusions

This paper proposed and analyzed a novel half-duplex two-way relaying (HD-TWR) framework incorporating digital network coding and rateless codes, with a focus on evaluating outage probability under different operating scenarios. The three-phase (SM3P) and four-phase (SM4P) schemes were investigated and closed-form OP expressions were derived, providing useful analytical tools for system optimization.

The simulation results demonstrate that the optimal relay position lies at the midpoint between the two sources, while the equal allocation of power between the nodes achieves the best outage performance.

Furthermore, increasing the maximum number of transmission attempts at the sources can enhance OP, though at the cost of higher latency and energy consumption. These findings

highlight the effectiveness of the proposed RC-DNC-based HD-TWR model in improving reliability and efficiency, while also providing practical guidelines for relay placement and resource allocation in cooperative wireless networks.

The proposed analysis is based on idealized assumptions with independent Rayleigh fading and equal retransmission probabilities. In practice, packet errors may be correlated, and the error rate may remain approximately constant within certain transmit power ranges due to hardware limitations.

However, the analytical trends derived here provide valuable qualitative information on how power allocation and relay placement affect system reliability. Despite these simplifications, the proposed model offers a tractable analytical foundation that can be extended to more realistic correlated-fading scenarios in future work.

For future work, the investigation of three-phase and four-phase HD-TWR models with advanced relay selection strategies, such as partial relay selection and optimal relay selection [7], [31], is suggested, as these approaches are expected to further enhance performance. Furthermore, the extension of the proposed framework to two-phase FD-TWR systems with RC-DNC integration will be considered to explore additional improvements in network performance.

Acknowledgments

This research is funded by University of Transport and Communications (UTC) under grant number T2025-PHII_DDT-001.

References

[1] M.N. Mowla *et al.*, "Internet of Things and Wireless Sensor Networks for Smart Agriculture Applications: A Survey", *IEEE Access*, vol. 11, pp. 145813–145852, 2023 (https://doi.org/10.1109/ACCESS.2023.3346299).

[2] M. Pundir *et al.*, "Dimensional-based Methods for Topological Management in Underwater Wireless Sensor Networks: A Comprehensive Survey", *IEEE Access*, vol. 13, pp. 67511–67530, 2025 (https://doi.org/10.1109/ACCESS.2025.3546978).

[3] M.F. Farsi *et al.*, "Deployment Techniques in Wireless Sensor Networks, Coverage and Connectivity: A Survey", *IEEE Access*, vol. 7, pp. 28940–28954, 2019 (https://doi.org/10.1109/ACCESS.2019.2902072).

[4] S. Najjar, M. David, W. Derigent, and A. Zouinkhi, "Dynamic Reconfiguration of Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 262, art. no. 111176, 2025 (https://doi.org/10.1016/j.comnet.2025.111176).

[5] A. John, I.F.B. Isnin, S.H.H. Madni, and M. Faheem, "Intrusion Detection in Cluster-based Wireless Sensor Networks: Current Issues, Opportunities and Future Research Directions", *IET Wireless Sensor Systems*, vol. 14, pp. 293–332, 2024 (https://doi.org/10.1049/wss2.12100).

[6] K. Loukil, "Energy Saving Multi-relay Technique for Wireless Sensor Networks Based on Hw/Sw MPSoC System", *IEEE Access*, vol. 11, pp. 27919–27927, 2023 (https://doi.org/10.1109/ACCESS.2023.3259235).

[7] M.S. Ghahroudi, A. Shahrabi, S.M. Ghoreyshi, and F.A. Alfouzan, "Distributed Node Deployment Algorithms in Mobile Wireless Sensor Networks: Survey and Challenges", *ACM Transactions on Sensor Networks*, vol. 19, pp. 1–26, 2023 (https://doi.org/10.1145/3579034).

[8] M.M. Moslehi, "Exploring Coverage and Security Challenges in Wireless Sensor Networks: A Survey", *Computer Networks*, vol. 260, art. no. 111096, 2025 (https://doi.org/10.1016/j.comnet.2025.111096).

[9] H. Zhang *et al.*, "Secure Resource Allocation for OFDMA Two-way Relay Wireless Sensor Networks without and with Cooperative Jamming", *IEEE Transactions on Industrial Informatics*, vol. 12, pp. 1714–1725, 2016 (https://doi.org/10.1109/TII.2015.2489610).

[10] Z. Iqbal, K. Kim, and H.-N. Lee, "A Cooperative Wireless Sensor Network for Indoor Industrial Monitoring", *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 482–491, 2017 (https://doi.org/10.1109/TII.2016.2613504).

[11] Aripriharta *et al.*, "A New Bio-inspired for Cooperative Data Transmission of IoT", *IEEE Access*, vol. 8, pp. 161884–161893, 2020 (https://doi.org/10.1109/ACCESS.2020.3021507).

[12] H.-H. Choi and K. Lee, "Cooperative Wireless Power Transfer for Lifetime Maximization in Wireless Multihop Networks", *IEEE Transactions on Vehicular Technology*, vol. 70, pp. 3984–3989, 2021 (https://doi.org/10.1109/TVT.2021.3068345).

[13] B. Hong and W. Choi, "Overcoming Half-duplex Loss in Multi-relay Networks: Multiple Relay Coded Cooperation for Optimal DMT", *IEEE Transactions on Communications*, vol. 63, pp. 66–78, 2015 (https://doi.org/10.1109/TCOMM.2014.2369054).

[14] C. Yao, H. Wu, and Z. Zhang, "A Novel Rateless Coded Protocol for Half-duplex Relaying Systems with Buffered Relay", *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*, Chicago, USA, 2018 (https://doi.org/10.1109/VTCFall.2018.8690562).

[15] S. Panic, D.N.K. Jayakody, and S. Garg, "Self-energized Bidirectional Sensor Networks over Hoyt Fading Channels under Hardware Impairments", *2019 IEEE 90th Vehicular Technology Conference (VTC2019-Fall)*, Honolulu, USA, 2019 (https://doi.org/10.1109/VTCFall.2019.8891415).

[16] H. Shen *et al.*, "Is Full-duplex Relaying More Energy Efficient Than Half-duplex Relaying?", *IEEE Wireless Communications Letters*, vol. 8, pp. 841–844, 2019 (https://doi.org/10.1109/LWC.2019.2895649).

[17] F.-K. Gong, J.-K. Zhang, and J.-H. Ge, "Asymptotic SEP Analysis of Two-way Relaying Networks with Distributed Alamouti Codes", *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 3777–3783, 2012 (https://doi.org/10.1109/TVT.2012.2210060).

[18] D. Jia *et al.*, "A Hybrid EF/DF Protocol with Rateless Coded Network Code for Two-way Relay Channels", *IEEE Transactions on Communications*, vol. 64, pp. 3133–3147, 2016 (https://doi.org/10.1109/TCOMM.2016.2583422).

[19] P. Popovski and H. Yomo, "Physical Network Coding in Two Way Wireless Relay Channels", *2007 IEEE International Conference on Communications*, Glasgow, UK, 2007 (https://doi.org/10.1109/ICC.2007.121).

[20] S. Jain and R. Bose, "Rateless Code-Aided Transmission Scheme to Achieve Secrecy in a Delay-Constraint Environment", *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Goa, India, 2019 (https://doi.org/10.1109/ANTS47819.2019.9118153).

[21] H. Wei and N. Deng, "On the Age of Information in Wireless Networks Using Rateless Codes", *IEEE Access*, vol. 8, pp. 173147–173157, 2020 (https://doi.org/10.1109/ACCESS.2020.3025431).

[22] D.-H. Ha *et al.*, "Security-reliability Trade-off Analysis for Rateless Codes-based Relaying Protocols Using NOMA, Cooperative Jamming and Partial Relay Selection", *IEEE Access*, vol. 9, pp. 131087–131108, 2021 (https://doi.org/10.1109/ACCESS.2021.3114343).

[23] H.-C. Lin, K.-H. Lin, and H.-Y. Wei, "Adaptive Age of Information Optimization in Rateless Coding-based Multicast-enabled Sensor Networks", *IEEE Journal of Selected Areas in Sensors*, vol. 1, pp. 73–92, 2024 (https://doi.org/10.1109/JSAS.2024.3407689).

[24] G. Huang *et al.*, "Improving Throughput in SWIPT-based Wireless Multirelay Networks with Relay Selection and Rateless Codes", *Digital Communications and Networks*, vol. 10, pp. 1131–1144, 2024 (https://doi.org/10.1016/j.dcan.2023.01.012).

[25] Y. Li *et al.*, "Relay Mode Selection and Power Allocation for Hybrid One-way/Two-way Half-duplex/Full-duplex Relaying", *IEEE Communications Letters*, vol. 19, pp. 1217–1220, 2015 (https://doi.org/10.1109/LCOMM.2015.2433260).

[26] J. Ma, C. Huang, and Q. Li, "Energy Efficiency of Full- and Half-duplex Decode-and-forward Relay Channels", *IEEE Internet of Things Journal*, vol. 9, pp. 9730–9748, 2022 (https://doi.org/10.1109/JIOT.2022.3143165).

[27] G. Srirutchataboon and S. Sugiura, "Secrecy Performance of Buffer-aided Hybrid Virtual Full-duplex and Half-duplex Relay Activation", *IEEE Open Journal of Vehicular Technology*, vol. 3, pp. 344–355, 2022 (https://doi.org/10.1109/OJVT.2022.3189612).

[28] L. Ong, "The Half-duplex Gaussian Two-way Relay Channel with Direct Links", *2015 IEEE International Symposium on Information Theory (ISIT)*, pp. 1891–1895, 2015 (https://doi.org/10.1109/ISIT.2015.7282784).

[29] A. Shokrollahi and M. Luby, "Raptor Codes", *Foundations and Trends in Communications and Information Theory*, vol. 6, pp. 213–322, 2011 (https://doi.org/10.1561/0100000060).

[30] P. Schulz *et al.*, "Efficient Reliable Wireless Communications through Raptor Codes and Rateless Codes with Feedback", *ICC 2022 - IEEE International Conference on Communications*, Seoul, South Korea, 2022 (https://doi.org/10.1109/ICC45855.2022.9838721).

[31] C. Huang *et al.*, "A Parallel Joint Optimized Relay Selection Protocol for Wake-up Radio Enabled WSNs", *Physical Communications*, vol. 47, art. no. 101320, 2021 (https://doi.org/10.1016/j.phycom.2021.101320).

The-Anh Ngo, Ph.D., Lecturer
Campus in Ho Chi Minh City
https://orcid.org/0009-0006-4976-1646
E-mail: anhnt_ph@utc.edu.vn
University of Transport and Communications, Ho Chi Minh City, Vietnam
https://utc2.edu.vn

Viet-Thanh Le, Lecturer
Faculty of Information Technology
https://orcid.org/0009-0000-8683-3158
E-mail: levietthanh@tdtu.edu.vn
Ton Duc Thang University, Ho Chi Minh City, Vietnam
https://tdtu.edu.vn

Thien P. Nguyen, Student
Faculty of Electrical and Electronics Engineering
https://orcid.org/0009-0008-0508-8167
E-mail: 42100891@student.tdtu.edu.vn
Ton Duc Thang University, Ho Chi Minh City, Vietnam
https://tdtu.edu.vn

Duy-Hung Ha, Ph.D., Lecturer
Wireless Communications Research Group
Faculty of Electrical and Electronics Engineering
https://orcid.org/0000-0001-6980-7273
E-mail: haduyhung@tdtu.edu.vn (Corresponding Author)
Ton Duc Thang University, Ho Chi Minh City, Vietnam
https://tdtu.edu.vn

AI-based Violent Incident Detection in Surveillance Videos to Enhance Public Safety

Khaled Merit and Mohammed Beladgham

Tahri Mohammed University of Bechar, Algeria

https://doi.org/10.26636/jtit.2025.4.2328

Abstract — Acts of violence may occur at any moment, even in densely populated areas, making it important to monitor human activities to ensure public safety. Although surveillance cameras are capable of detecting the activity of people, around-the-clock monitoring still requires human support. As such, an automated framework capable of detecting violence, issuing early alerts, and facilitating quick reactions is required. However, automation of the entire process is challenging due to issues such as low video resolution and blind spots. This study focuses on detecting acts of violence using three video data sets (movies, hockey game and crowd) by applying and comparing advanced ResNet architectures (ResNet50V2, ResNet101V2, ResNet152V2) with the use of the bidirectional gated recurrent unit (BiGRU) algorithm. Spatial features of each video frame sequence are extracted using these pre-trained deep transfer learning models and classified by means of an optimized BiGRU model. The experimental results were then compared with those achieved by wavelet feature extraction approaches and other classification models, including CNN and LSTM. Such an analysis indicates that the combination of ResNet152V2 and BiGRU offers decent performance in terms of higher accuracy, recall, precision, and F1 score across the different datasets. Furthermore, the results indicate that deeper ResNet models significantly improve overall performance of the model in terms of violence detection scores, relative to shallower ResNet models. ResNet152V2 was found to be the ultimate model across the datasets when it comes to a high degree of accuracy in detecting acts of violence.

Keywords — bidirectional gated recurrent unit, deep learning, deep transfer learning, video processing, violence detection

1. Introduction

Violence is defined as deliberate exertion of physical force intended to harm, dominate, or manipulate individuals or groups. Actions of this type are considered to constitute criminal conduct that transgresses legal boundaries, societal expectations, and moral principles followed by global communities. The effects of violence are multifaceted, encompassing not only bodily injury, but also deep emotional and psychological distress which can, in extreme situations, lead to fatal outcomes.

Currently, the use of closed-circuit television (CCTV) is increasing because the solution is capable of providing non-stop surveillance – a task humans cannot accomplish. Cameras record all the events from various angles. As a result, a large amount of video data still requires a human to identify unwelcome types of activity, including violence. If performed manually, this video-monitoring process requires significant amounts of time and effort. Therefore, it is necessary to employ an automatic detection system that will accelerate the entire procedure. One of the challenges faced when performing automatic detection is low resolution of the video feed generated by CCTV cameras [1] (resulting from poor lighting, ambient conditions, distance, and hardware constraints).

Automatic event detection has been possible for several years now, and the process of detecting acts of violence is similar to that of recognizing actions [2]. The difference is that violence detection focuses not only on movement, but also on the intention of that movement. In this case, the speed of movement that occurs will determine whether a given action is categorized as an act of violence or just an ordinary movement.

The authors of [3]–[8] detect objects in CCTV video data. However, not all acts of violence, such as hand-to-hand fights or altercations, involve weapons. Therefore, it is necessary to detect acts of violence that do not depend on a suspicious object.

Several studies have been conducted that focused on detecting acts violence, with various approaches relied upon in the process. The author of [9] used a histogram of optical flow (HOF) to extract valuable features from videos, while in [10], HOF magnitude and orientation (HOMO) are used. In [11], motion features are extracted from dynamic RGB images, while in [12] convolutional neural network (CNN) models (namely VGG-19, ResNet50 and Xception) are employed, with each of them trained using the ImageNet dataset. The results they achieve are reasonably good.

Studies [13] and [14] used VGG-16 for feature extraction and a simple SVM classification algorithm. Better results were obtained in 0, which used ResNet50 as the backbone for three-dimensional CNNs and dense optical flow for the region of interest.

Another difficulty encountered while detecting acts of violence with the use of surveillance cameras stems from the presence of crowds in public places. The violent flow dataset, also known as the crowd dataset, is one example of a dataset containing videos of public crowds. Several studies have re-

lied upon the crowd data set. For example, the author of [16] used a violent flow (ViF) descriptor and then classified the output using a linear SVM, achieving a precision score of 81.3%.

Using the same classification algorithm combined with HOF, the researchers in [17] obtained an accuracy of 83.37%. That result still needs to be improved to create a precise detection system. This dataset is challenging because acts of violence are sometimes not visible due to the density of the crowd.

On the other hand, crowded conditions often lead to false positives. Therefore, in this study, acts of violence were detected using the crowd dataset, with the overall aim of improving the quality of the model in terms of its performance and lead time.

To classify data into appropriate classes, a powerful classification model is needed. the following solutions were used: bidirectional gated recurrent unit model (BiGRU), long-short-term memory model (LSTM), CNN, etc. The LSTM model used images and also accepted other data types, such as text, and achieved very good accuracy levels [18].

The main contribution of this work is a benchmark of advanced ResNet architectures (ResNet50V2, ResNet101V2, ResNet152V2) against classical and other deep learning-based feature extractors, when combined with various temporal classifiers (CNN, LSTM, BiGRU). Our work provides a clear evidence-based pathway for selecting model components, demonstrating that the synergistic combination of ResNet152V2 and BiGRU delivers superior and consistent state-of-the-art performance across diverse benchmark datasets.

We used ResNet50V2, ResNet101V2, and ResNet152V2 to extract vital features from video and wavelets. BiGRU was selected as a classification algorithm, as it offers better results than LSTM in terms of predicting the condition of a pulp paper press [19]. In the classification of emotions in noisy speech, BiGRU provides a shorter run time and a lower error rate while removing noise, compared to LSTM [20]. For comparison, this research also uses the LSTM and CNN algorithms.

The structure of this paper is as follows. The description and video pre-processing stages are detailed in Section 2. The methods used for extracting violence-specific features are explained in Section 3. Violence classification algorithms are presented in Section 4. Experimental results and discussion, including computational efficiency analysis, are provided in Section 5. Ethical considerations are discussed in Section 6. Conclusions and future work are described in Section 7.

2. Video Detection

A general scheme for detecting violent acts is illustrated in Fig. 1. Initially, it is essential to pre-process the video data, followed by a systematic categorization into training and testing datasets utilizing k-fold validation. Subsequently, the feature extraction stage is performed using ResNet50V2, ResNet101V2, and ResNet152V2. We also compared the features extracted using several methods: principal component analysis (PCA), discrete wavelet transforms (DWT), VGG-16 and VGG-19.

The most effective method of extracting features from the training data were used to develop a violence detection model employing the BiGRU algorithm. We compared the classification model with several algorithms such as CNN and LSTM. In the final stage, the model was assessed using the test dataset. This evaluation utilized metrics such as accuracy, recall, specificity, G-mean, and CPU time to thoroughly gauge the effectiveness of the model.

In addition to ResNetXV2, we also compared wavelet feature extraction methods and non-feature extraction to compare performance in terms of violence detection and extraction processing time.

2.1. Dataset

Data from three datasets were used in this research to assess the performance of the model in detecting violence in a video: movies [21], hockey game [21], and crowd [16] (Tab. 1).

The videos in the movies dataset contain several movie scenes and consist of 200 clips divided into 100 fight and 100 non-fight sequences. The hockey dataset contains 1000 video recordings of matches from the National Hockey League, divided into 500 violent and 500 non-violent clips. The crowd dataset is a real-time video recording of violence in a crowd, containing 246 videos with 123 violent and 123 non-violent clips. Each dataset was divided into training and test datasets using k-fold validation.
Figures 2, 3 present sample frames from each dataset.

2.2. Pre-processing

Pre-processing phase is the preliminary step in building a violence-detection system. In this step, each video is converted into a series of RGB format images. These images are subsequently resized to 224 × 224 pixels to align with the input specifications of the ResNet models.

The next phase involves extracting the pixel intensities from each set of images. In this scenario, we obtained a matrix with dimensions $m \times n \times 224 \times 224 \times 3$. In this scenario, m signifies the total number of clips, n indicates the number of images captured per recording session, and $224 \times 224 \times 3$ specifies the dimensions of an RGB image in bytes.

Tab. 1. Brief description of datasets used to detect violence in video footage.

Datasets	Frame size	No. of clips	Violence	No violence	Format
Movies [21]	576 × 720	200	100	100	.mpg .mp4
Hockey [21]	288 × 360	1000	500	500	.avi
Crowd [16]	240 × 320	246	123	123	.avi

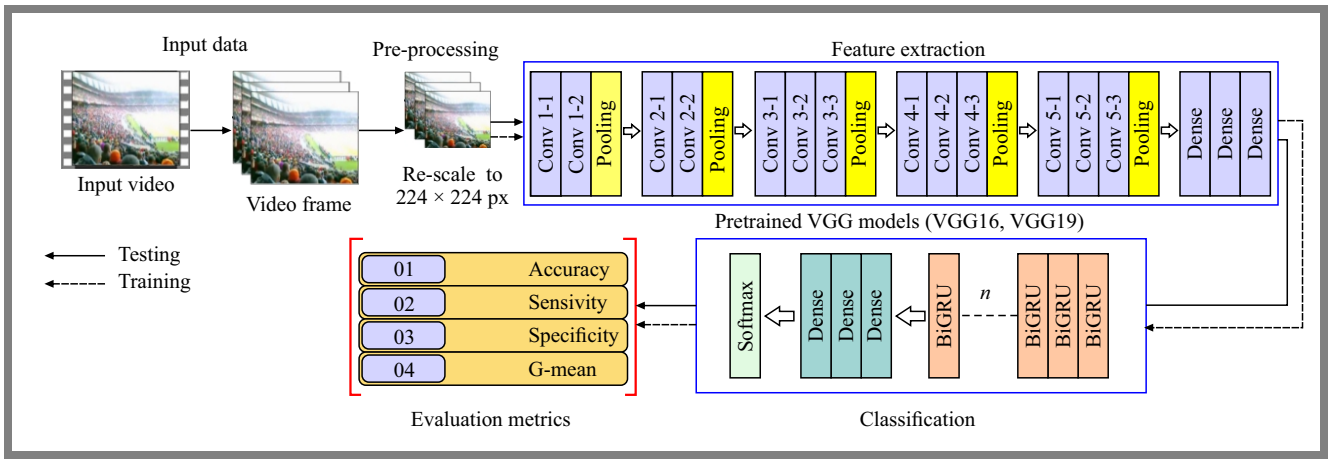

Fig. 1. Schematic of the violence detection system.

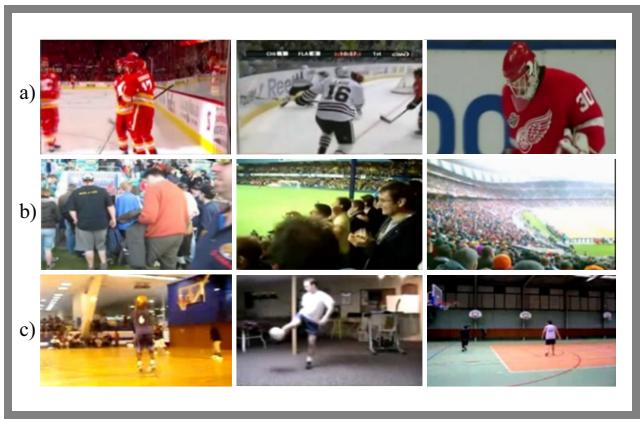

Fig. 2. Sample frames of non-violent video benchmark datasets: a) hockey dataset, b) crowd dataset, and c) movie dataset.

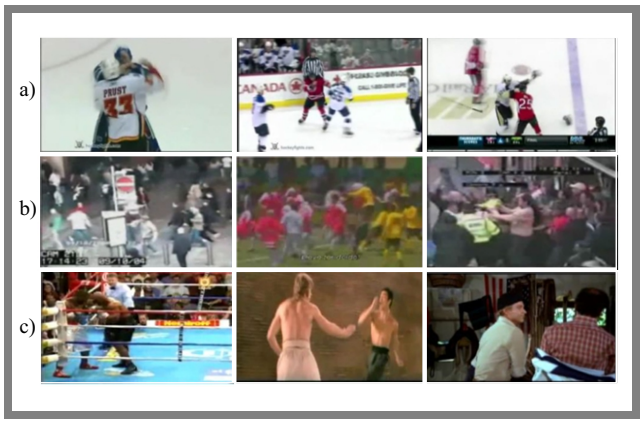

Fig. 3. Sample frames of violent video benchmark datasets: a) hockey dataset, b) crowd dataset, and c) movie dataset.

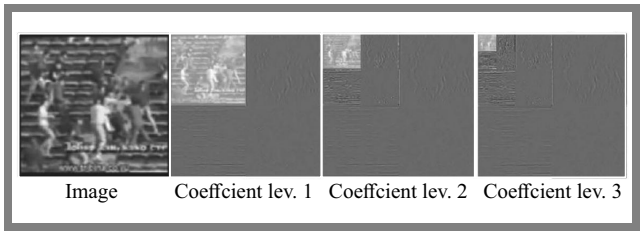

Fig. 4. Feature extraction process using DWT.

3. Violence Feature Extraction

3.1. Discrete Wavelet Transform

In this research, level 3 wavelet decomposition was used to compare the feature extraction methods. The mother wavelet uses Daubechies 8, N (in Db), where N represents the Daubechies polynomial order. The wavelet of the Daubechies order $N \geqslant 2$ has $2N$ vanishing moments and a small-scale support with an interval of $[0, 2N-1]$ [22]. The Daubechies polynomial order $N-1$ is defined as:

$$P_{N-1}(y) = \sum_{k=0}^{N-1} \binom{2N-1}{k} y^k (1-my)^{N-1-k} . \qquad (1)$$

After obtaining a grayscale image, level 1 wavelet decomposition is performed and then LL, LH, HL, and HH sub-bands are obtained. The LL sub-band contains the approximate value of the image and is the input for the next decomposition level. The sub-band used during the classification process is the approximate value of the level 3 wavelet decomposition. In this process, a matrix measuring $m \times n \times 41 \times 41$ is produced. The matrix is reshaped to adjust the input dimensions in the classification process. The results of feature extraction using the DWT are shown in Fig. 4.

3.2. Principal Component Analysis

Principal component analysis (PCA) is a transformation technique that converts and decomposes a large set of correlated variables into a smaller set of uncorrelated variables. This method effectively reduces the dimensionality of the data while preserving essential information. Each image frame is converted to grayscale and dimensioned into a row vector with dimension $(1 \times m)$, where m is $n \times n$, and n is the size of the image. For each dataset, all vectors were aggregated into a size matrix of size $(N \times 50\,176)$, where N is the number of images. The next step is to select the value of the principal component with k percent of the total eigenvalues. The results of the feature extraction using PCA are shown in Fig. 5.

3.3. Residual Networks (ResNet)

ResNet is a deep learning approach and is an evolution of a CNN. In the learning process, ResNet implements residual connections that can connect layers to other layers by skipping some middle layers. It is claimed to avoid the vanishing gradient problems that occur during the training process [23]. More than the use of a deep learning architecture alone is needed to increase the accuracy of the learning process. Therefore, to improve recognition accuracy, transfer learning is used.

Transfer learning is an approach to deep learning (and machine learning) in which knowledge is transferred from one model to another. A common misconception regarding transfer learning is that training and test datasets must come from the same source or have the same distribution. In practice, however, the transferred tasks may differ in the same domain. In common deep neural networks, models learn only from existing data. With limited data, it will be difficult for the model to obtain optimal recognition results. Deep transfer learning, on the contrary, using pre-trained models trained on other datasets in the same domain, can boost classification performance [24].

ResNet50V2, ResNet101V2, and ResNet152V2 are improved versions of their respective ResNet families, incorporating identity mapping and applying batch normalization and ReLU activation before the weight layers, which enhances gradient flow and training stability [25]. The key characteristics of these architectures, which form the basis of our feature extraction comparison, are summarized in Tab. 2.

The ResNetXV2 architecture is illustrated in Fig. 6. In this study, we used these models as feature extractors for the input video. Subsequently, we advanced the learning process by employing additional deep learning techniques, including CNN, LSTM, and BiGRU, as classification methods. The matrix resulting from block 5 for each ResNet variant is characterized by dimensions of $m \times n \times 7 \times 7 \times 512$.

The matrix is subsequently processed through the flatten and dense layers, resulting in a matrix of size of $m \times n \times 4096$. The characteristic extraction procedure utilizing ResNet ends at the dense layer and further processing is conducted using an alternative classifier.

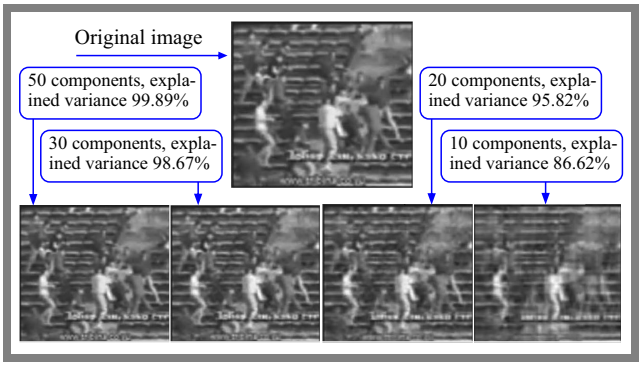

Fig. 5. Feature extraction process using PCA.

Tab. 2. Architectural comparison of the ResNetV2 models used to extract features.

Model	Depth (layers)	Parameters (millions)	Bottleneck blocks	Complexity
ResNet50V2	50	25.6	16 3×[3, 4, 6, 3]	Medium
ResNet101V2	101	44.6	33 3×[3, 4, 23, 3]	High
ResNet152V2	152	60.2	50 3×[3, 8, 36, 3]	Highest

3.4. VGG

VGG represents a convolutional neural network (CNN) framework that was developed using the ImageNet database [24]. VGG can handle massive datasets, as it contains several weighted layers with millions of parameters. The difference between VGG-16 and VGG-19 networks is the depth of the weight layers, as shown in Fig. 7. In VGG-16, the number of weight layers is 16, whereas VGG-19 has a layer depth of 19. We used VGG-16 and VGG-19 as a comparison feature extractor for the input video. The output matrix from block 5 of the VGG-16 model has dimensions of $m \times n \times 7 \times 7 \times 512$. After being processed through both the flatten and dense layers, the matrix is reconfigured to have dimensions of $m \times n \times 4096$.

4. Violance Classification

After acquiring the feature set from the trained ResNet models, we compared several deep learning methods to detect acts of violence in a given. The CNN in this study consists of three convolution and max-pooling layers. The CNN architecture is shown in Fig. 8a. The hyperparameter settings for the CNN were an initial learning rate of 0.1, a batch size of 100, 200 epochs, a dense kernel size of 100, a loss function based on mean squared error, and SGD optimizers.

LSTM is an advanced recurrent neural network that solves the vanishing gradient problem [26]. Each LSTM cell has three gates, namely a forget gate, an input gate, and an output gate (Fig. 8b). In this study, the hyperparameter settings for LSTM were as follows: an initial learning rate of 0.1, a batch size of 100, 100 epochs, a dense kernel size of 100, a loss function based on mean squared error, and the Adam optimizer.

GRU was introduced in [27], with its design similar to that of the LSTM but using a more straightforward memory unit to simplify training and implementation. In this study, classification was performed using a bidirectional GRU (BiGRU) to better capture contextual information from both past and future frames (Fig. 8c). The result of the feature extraction stage using ResNet passes through the BiGRU layer in this process. Furthermore, the resulting BiGRU matrix goes through three dense layers and the last output goes through a dense layer with two units using the softmax activation function. This layer maps the classification results into two class labels: violence or non-violence. The hyperparameter settings for the BiGRU were an initial learning rate of 0.1, a batch size of 100, 100 epochs, a dense kernel size of 100, a loss function based on mean squared error and the Adam optimizer.

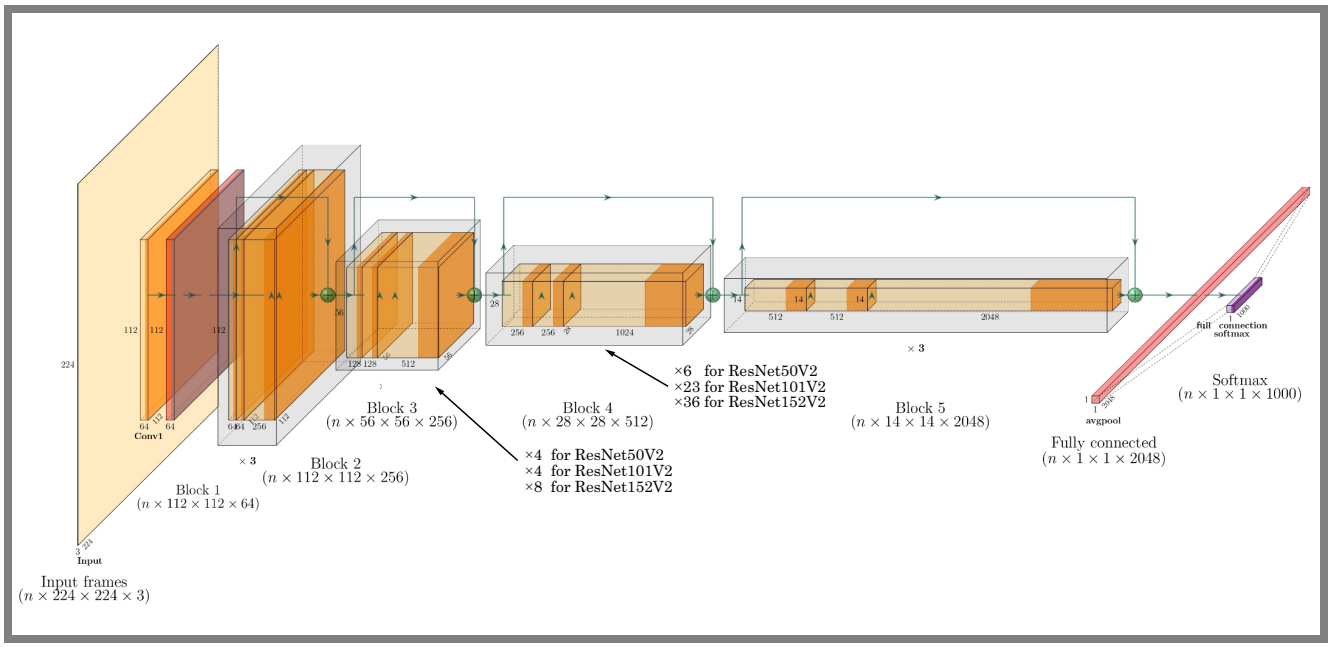

Fig. 6. Three-dimensional ResNetXV2 architecture.

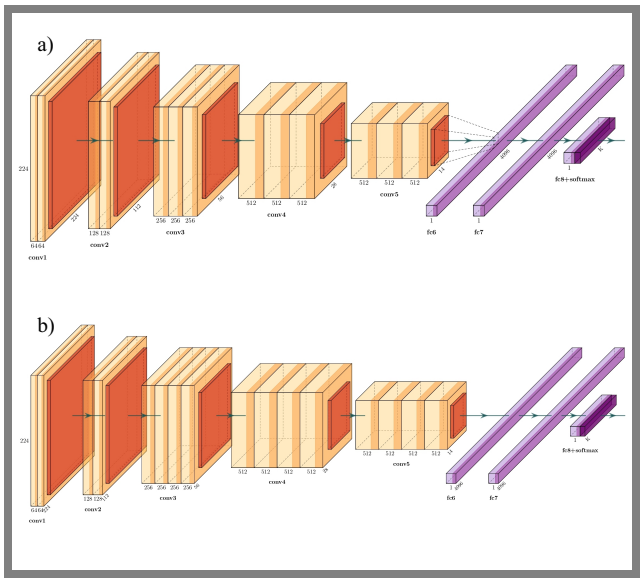

Fig. 7. VGG architecture: a) VGG16 and b) VGG19.

The bidirectional architecture enables the model to capture more comprehensive temporal patterns, which is particularly beneficial for violence detection in video sequences, where context from both preceding and subsequent frames is crucial for accurate classification.

5. Results and Discussion

In this study, acts of violence were classified using the following publicly available datasets: movies, hockey game, and crowd. We used a deep transfer learning approach based on ResNet50V2, ResNet101V2, and ResNet152V2 to extract essential features from the data. Furthermore, we compared the experimental results using Daubechies-8 wavelet and PCA as classical feature extraction methods, and VGG-16 and VGG-19 as deep transfer learning-based feature extraction methods. The pre-trained weights obtained from the ImageNet dataset were used, since the images in ImageNet have a resolution of 224 × 224, which matches the CCTV image frame input. Additionally, ImageNet has approximately 14 million images grouped into 1000 various categories. The use of a model pre-trained on ImageNet certainly improves learning outcomes on violence datasets and ensures good recognition performance.

We divided the training and test data using 10-fold cross-validation. CNN, LSTM, and BiGRU classification algorithms were used. The parameters used for the evaluation of the model included the following: accuracy, recall, precision, and F1 score. We also considered performance of the model in terms of the time required for feature extraction, training, and testing for each dataset. The experimental results are listed in Tabs. 3, 4, and 5.

Table 4 indicates that a combination of ResNet152V2 and BiGRU produces the maximum accuracy of 1.000 in the hockey dataset. In addition to achieving the highest degree of precision, the ResNet152V2-BiGRU combination produced the best precision, recall and F1 score values of 1.000 in each metric. This shows that the model's ability to classify the two classes is better than that of other algorithm combinations.

As in the hockey dataset, the best accuracy on the crowd data set (1.000) is also obtained when the ResNet152V2-BiGRU combination is used (Tab. 5). If reviewed further, the use of ResNet152V2 for feature extraction improved model performance, as evidenced by the increase in precision, recall, precision, and F1 score, compared with classical and older feature extraction approaches.

However, using deep transfer learning features yields significantly better results when compared with classical feature

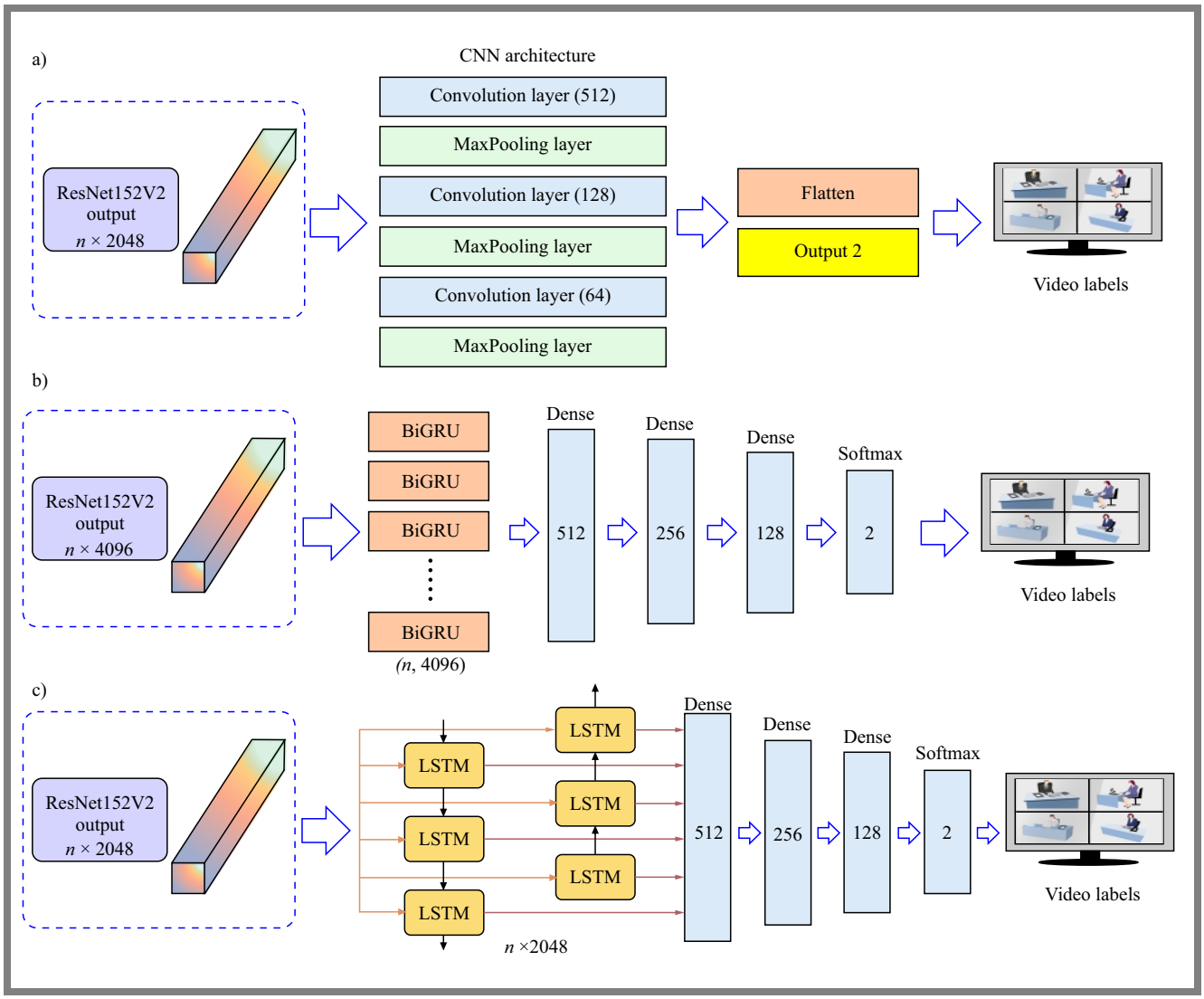

Fig. 8. a) CNN, b) BiGRU, and c) LSTM architectures.

extraction. It can be found that the model built with the crowd dataset using the ResNet152V2-BiGRU combination obtained the best performance, as it achieves the best accuracy and obtains the best metric results (all 1.000).

In contrast to the previous two datasets, the experimental results on the movie dataset were 1.000 for most classification methods and metrics. These excellent metric scores were achieved by all combinations of algorithms, except for BiGRU and its combination with Daubechies-8 wavelets and PCA.

This occurrence can be attributed to the fact that the video in this dataset represents a particular instance of a film scene, where the lighting and camera angles have been deliberately configured. Therefore, the video is clear and does not contain much noise. This differs from the hockey and crowd datasets, which were obtained from surveillance cameras.

Tables 3 – 5 also present the time required to perform feature extraction on a data set and the classification time. One may notice that feature extraction using ResNet152V2 takes longer, but for the training process, ResNet152V2 is faster than VGG-16 and VGG-19. Table 3 also presents the CPU time required to process one test video. The fastest time was obtained using VGG-19. For the crowd data set, ResNet152V2-based feature extraction improves model performance. However, this also increases the time required to process the test data. Upon further analysis, for an increase in accuracy of up to 0.25, a time difference of 0.1 to 0.6 s can be tolerated.

Furthermore, one of the advantages of BiGRU is that in terms of time, it is faster than LSTM, as fewer parameters are used in BiGRU. Consequently, BiGRU is more efficient in terms of memory and time. The results show that BiGRU can perform successfully on all datasets in this study.

Although the proposed ResNet152V2-BiGRU model achieved perfect evaluation metrics (1.000) on the hockey and crowd datasets, it is important to contextualize these results. Its good performance can be attributed to the model's strong capacity for spatio-temporal feature learning on these specific benchmarks. We employed a 10-fold cross-validation strategy to minimize the risk of overfitting and data leakage, and the convergence of training and validation curves (Fig. 9) supports the model's generalization within these datasets.

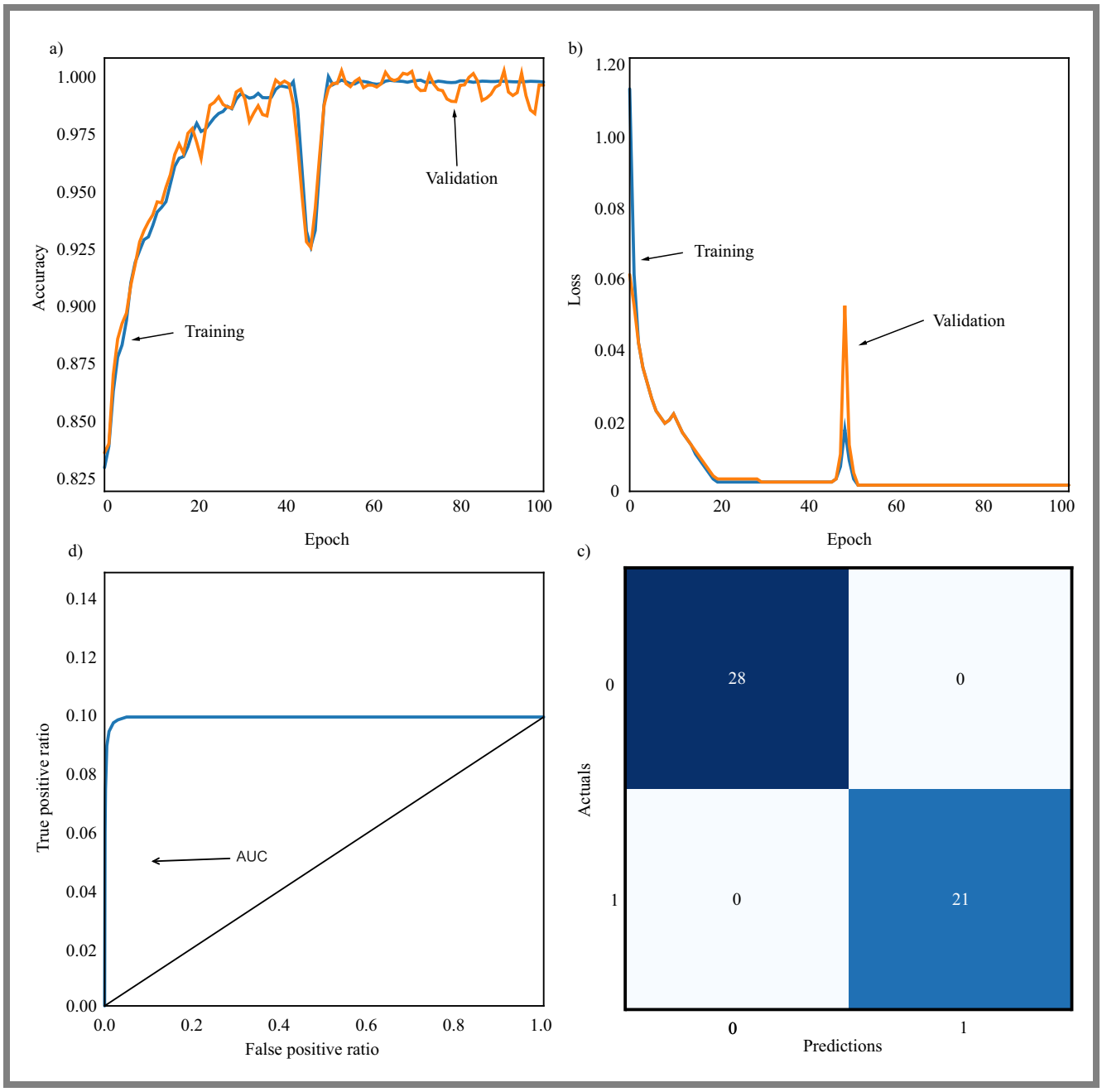

Fig. 9. Performance evaluation of the ResNet152V2 model: a) training accuracy, b) model loss, c) confusion matrix, and d) receiver operating characteristic (ROC) curve.

However, these results, while indicative of a high degree of effectiveness, should be interpreted with the understanding that real-world surveillance footage poses additional unmodeled challenges. The following section discusses computational trade-offs and the need for future validation on larger, more complex real-world streams. The perfect scores show that the ResNet152V2+ BiGRU model can learn optimally on a violent data set to recognize the patterns in each category very effectively.

Figure 9 shows the accuracy and loss results during training and validation. It can be seen that the performance of the model decreased at the 50th epoch but stabilized by the 100th epoch and did not experience overfitting when the results between training and validation almost overlapped and were not significantly different.

In addition, we compared the accuracy of the proposed method with other studies that also used data sets from movies, hockey games, and crowds. Violent event detection using deep transfer learning provides excellent recognition, and almost all models obtained perfect evaluation metrics.

However, not all classifier models correctly detect every relevant class. In Fig. 10, we present a scatter plot of the recognition results for each data instance in the crowd dataset.

Tables 3 – 5 reveal that the best recognition results were obtained using the ResNet152V2 transfer learning model and

Tab. 3. Experimental feature extraction results on the movies datasets – hockey.

Classifier	Feature extraction	Extraction time [s]	Training time [s]	Testing time [s]	Accuracy	Recall	Precision	F1 score
LSTM	PCA	0.043	12.046	0.522	0.810	0.764	0.862	0.811
	Wavelet	0.013	11.514	0.582	0.940	0.915	0.968	0.941
	VGG-16	0.112	43.465	0.536	0.950	0.950	0.951	0.950
	VGG-19	0.106	27.753	0.438	0.970	0.970	0.970	0.970
	ResNet50V2	0.231	22.164	0.415	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	22.300	0.420	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	22.500	0.425	1.000	1.000	1.000	1.000
BiGRU	PCA	0.043	2.079	0.264	0.755	0.806	0.708	0.756
	Wavelet	0.013	3.266	0.900	0.865	0.783	0.957	0.866
	VGG-16	0.112	27.863	0.873	0.975	0.975	0.975	0.975
	VGG-19	0.106	26.900	0.388	0.965	0.965	0.965	0.965
	ResNet50V2	0.231	22.430	0.697	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	22.600	0.700	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	22.800	0.705	1.000	1.000	1.000	1.000
CNN	PCA	0.043	3.700	0.438	0.810	0.886	0.736	0.811
	Wavelet	0.013	15.453	1.060	0.945	0.934	0.957	0.946
	VGG-16	0.112	262.702	0.885	0.915	0.915	0.915	0.915
	VGG-19	0.106	263.022	0.318	0.900	0.900	0.901	0.900
	ResNet50V2	0.231	142.571	0.751	0.990	0.990	0.990	0.990
	ResNet101V2	0.378	143.000	0.760	0.995	0.995	0.995	0.995
	ResNet152V2	0.492	144.000	0.770	1.000	1.000	1.000	1.000

Tab. 4. Experimental feature extraction results on the movies datasets.

Classifier	Feature extraction	Extraction time [s]	Training time [s]	Testing time [s]	Accuracy	Recall	Precision	F1 score
LSTM	PCA	0.043	2.525	0.535	0.825	0.882	0.750	0.822
	Wavelet	0.013	2.904	0.459	1.000	1.000	1.000	1.000
	VGG-16	0.112	13.067	0.592	1.000	1.000	1.000	1.000
	VGG-19	0.106	12.243	0.423	1.000	1.000	1.000	1.000
	ResNet50V2	0.231	12.050	0.429	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	12.180	0.435	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	12.350	0.440	1.000	1.000	1.000	1.000
BiGRU	PCA	0.043	5.178	0.530	0.825	0.842	0.800	0.825
	Wavelet	0.013	5.484	0.440	0.975	0.950	1.000	0.975
	VGG-16	0.112	22.319	0.526	1.000	1.000	1.000	1.000
	VGG-19	0.106	13.120	0.362	1.000	1.000	1.000	1.000
	ResNet50V2	0.231	10.503	0.394	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	10.650	0.402	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	10.800	0.408	1.000	1.000	1.000	1.000
CNN	PCA	0.043	3.324	0.267	1.000	1.000	1.000	1.000
	Wavelet	0.013	4.519	0.902	1.000	1.000	1.000	1.000
	VGG-16	0.112	49.161	0.225	1.000	1.000	1.000	1.000
	VGG-19	0.106	82.547	0.157	1.000	1.000	1.000	1.000
	ResNet50V2	0.231	30.721	0.540	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	31.000	0.550	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	31.500	0.560	1.000	1.000	1.000	1.000

Tab. 5. Experimental feature extraction results on the movies datasets – crowd.

Classifier	Feature extraction	Extraction time [s]	Training time [s]	Testing time [s]	Accuracy	Recall	Precision	F1 score
LSTM	PCA	0.043	3.438	0.554	0.490	0.474	0.375	0.474
	Wavelet	0.013	2.797	0.526	0.625	0.583	0.667	0.624
	VGG-16	0.112	23.860	0.435	0.980	0.980	0.981	0.980
	VGG-19	0.106	23.441	0.402	0.939	0.939	0.946	0.939
	ResNet50V2	0.231	11.406	0.757	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	11.550	0.760	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	11.700	0.765	1.000	1.000	1.000	1.000
BiGRU	PCA	0.043	2.120	0.522	0.500	0.500	0.250	0.433
	Wavelet	0.013	2.537	0.511	0.656	0.690	0.625	0.657
	VGG-16	0.112	14.332	0.360	1.000	1.000	1.000	1.000
	VGG-19	0.106	22.837	0.360	1.000	1.000	1.000	1.000
	ResNet50V2	0.231	11.975	0.368	1.000	1.000	1.000	1.000
	ResNet101V2	0.378	12.100	0.375	1.000	1.000	1.000	1.000
	ResNet152V2	0.492	12.250	0.380	1.000	1.000	1.000	1.000
CNN	PCA	0.043	3.302	0.355	0.592	0.563	0.750	0.574
	Wavelet	0.013	3.752	0.329	0.667	0.750	0.583	0.661
	VGG-16	0.112	82.546	1.384	0.898	0.898	0.915	0.897
	VGG-19	0.106	57.631	0.164	0.694	0.694	0.691	0.690
	ResNet50V2	0.231	41.594	0.937	0.980	0.980	0.980	0.980
	ResNet101V2	0.378	42.000	0.945	0.985	0.985	0.985	0.985
	ResNet152V2	0.492	42.500	0.950	1.000	1.000	1.000	1.000

recognition comparisons were performed using the BiGRU, LSTM, and CNN models. On the 49th test data, four were miss-classified when the CNN+ ResNet152V2 model was used, while neither the BiGRU+ ResNet152V2 model nor the LSTM+ ResNet152V2 model output any misclassifications.

Figure 11 shows the detection results for each video in the video test data, by including the probability of recognizing violence and non-violence. The recognition results show the prediction results of the BiGRU+ ResNet152V2 combination, which is the best of the compared models. This model was then tested in the crowd, movies, and hockey datasets. Each image in the left column has a ground truth class of "violence" and each image in the right column has a ground truth class of "non-violence". The prediction results for each video show that the detection results are the same as the ground truth, with a high confidence rate for each class.

5.1. Computational Efficiency and Real-time Feasibility

Computational efficiency is a critical consideration for the deployment of AI models in real world systems. As shown in Tab. 3, there is a clear trade-off between model performance and processing time. Although ResNet152V2 has the longest feature extraction time (0.492 s per image), it yields the highest accuracy. To assess real-time feasibility, we consider the processing time per video clip. For the crowd dataset, the total test time for the ResNet152V2-BiGRU model was 0.38 s per video. Assuming a standard video clip length of a few seconds, this demonstrates good potential for near-real-time analysis in a processed clip-based system.

However, for true real-time streaming at standard frame rates (e.g., 25 – 30 fps), the current model requires optimization. Future work will focus on employing more efficient feature extractors (e.g., MobileNet, EfficientNet) and model compression techniques (e.g., pruning, quantization) to bridge this gap without a significant sacrifice in accuracy. BiGRU's faster training and testing time compared to LSTM, due to its simpler gating mechanism, is a positive step towards this goal.

In terms of the complexity and time consumption of the proposed model, it can be seen in Tabs. 3 – 5 that each deep-transfer learning model has a different extraction time. The longest feature extraction time was obtained using ResNet152V2 with an execution time of 0.492 s for each image, while the fastest feature extraction execution time was achieved for the wavelet, with an execution time of 0.013 s. ResNet152V2 has the longest extraction time, where the transfer learning process is quite complex because it uses many residual networks, causing the learning process to take longer than in the case of other transfer learning models.

The longest training process was that of CNN with VGG-19 feature extraction (263.022 s). A comparison with other studies is presented in Tab. 6. In the movies dataset, the proposed method outperforms the other methods with the highest scoring accuracy of 100%.

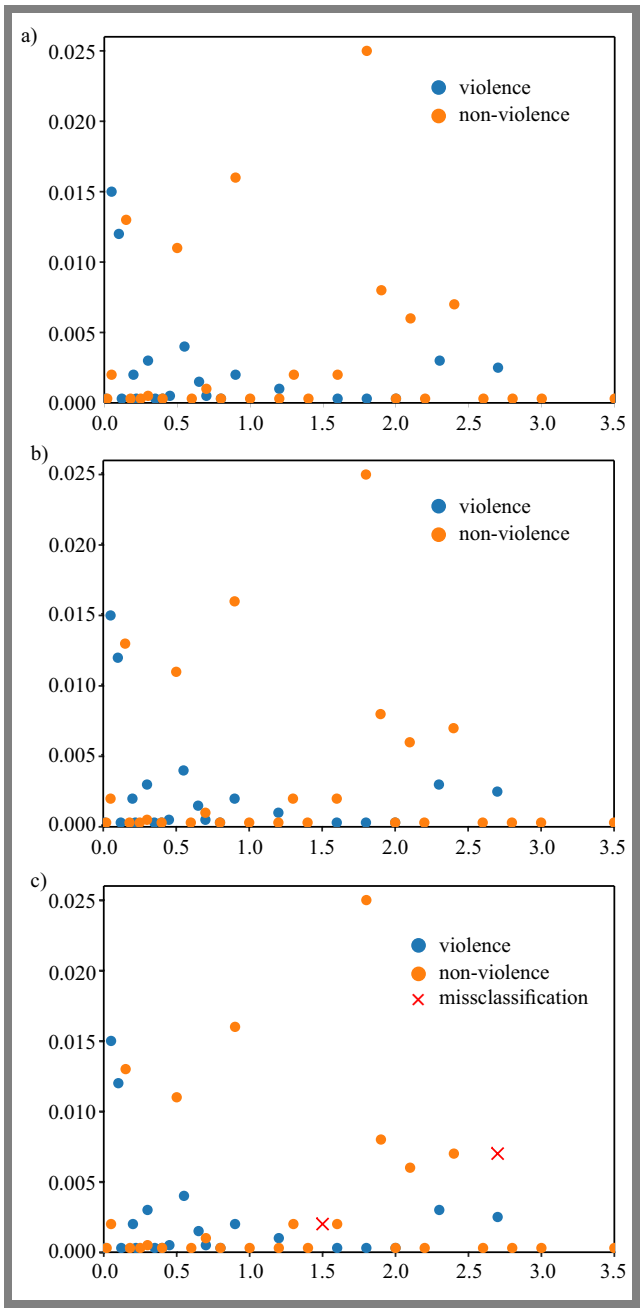

Fig. 10. Scatter plot of: a) BiGRU + ResNet152V2, b) LSTM + ResNet152V2, and c) CNN +ResNet152V2.

6. Ethical Considerations

The deployment of AI-based violence detection systems in public spaces requires a serious discussion focusing on ethical implications. Continuous video monitoring and analysis inherently raise privacy concerns. It is imperative that such systems are deployed in compliance with data protection regulations (e.g., GDPR). Strategies such as on-edge processing, where video data is analyzed locally without being stored or transmitted, can help mitigate privacy risks.

AI models can perpetuate and amplify societal biases if trained on non-representative data. Future work must include rigorous bias auditing across different demographics to ensure that the model does not disproportionately target specific groups.

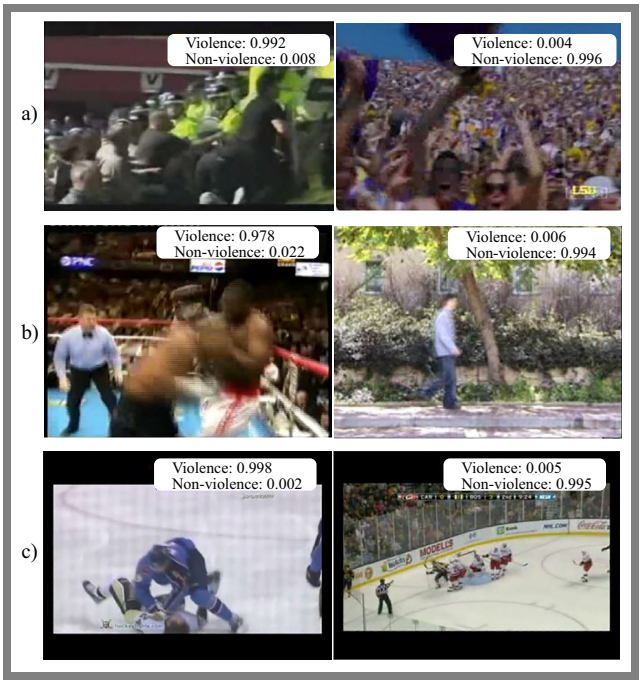

Fig. 11. Detection performance of BiGRU + ResNet152V2 in the: a) crowd, b) movies, and c) hockey datasets.

A false positive, where a non-violent act is flagged as violent, could lead to unnecessary alarm, wasted security resources, and potentially serious confrontations. Therefore, achieving high precision is not just a technical goal but an ethical imperative. In practice, such systems should function as an assistive tool for human operators who make the final decision, rather than as a fully autonomous response trigger.

Transparency in system capabilities and limitations, along with clear governance frameworks, is essential for the responsible development and deployment of this technology.

7. Conclusions

In this study, a comparative analysis of various solutions capable of detecting acts of violence in videos was conducted. The key finding is that the combination of ResNet152V2 for spatial feature extraction and BiGRU for temporal modeling represents a highly effective and efficient architecture, as validated by its top-tier performance with the use of the movies, hockey, and crowd data sets. ResNet50V2, ResNet101V2 and ResNet152V2 were used for feature extraction, while classical (wavelet and PCA), and other deep transfer learning methods (VGG-16 and VGG-19) were used as comparison methods.

Furthermore, CNN, LSTM, and BiGRU algorithms were used for classification. The best precision results in the hockey dataset were obtained when using the ResNet152V2-BiGRU combination. Furthermore, in the movies dataset, all combinations of algorithms achieved excellent performance (1.000). Similarly to the hockey dataset, the best accuracy on the crowd data set was achieved using the ResNet152V2-BiGRU combination. Furthermore, ResNet152V2-BiGRU provides the best accuracy, recall, precision, and F1 score performance.

Tab. 6. Comparison of the proposed violence detection system with state-of-the-art approaches.

		Accuracy [%]		
Ref.	Method	Movies	Hockey	Crowd
[10]	HOMO	–	89.3	76.8
[15]	Violence4D (4D-CNN)	100	100	97.29
[16]	ViF	96.7	81.6	81.2
[28]	MoWLD + BoW	–	91.9	82.5
	MoWLD + SparseCoding	–	93.7	86.3
[29]	ConFeat	96.5	94.4	80.9
[30]	sHOT	–	–	82.9
[31]	DIMOLIF	–	88.6	85.8
[32]	LHOF + BoW	–	–	86.5
[33]	BoW (MoBSIFT) + MF	98.9	90.3	–
[34]	AlexNet + 3D-CNN	98.7	92.9	88.0
[35]	Xception + LSTM	–	91	88
	InceptionV3 + LSTM	–	90	89
[36]	OViF	–	84.2	76.8
[37]	3D CNN + interest frames	100	99.4	97.49
[38]	Hough forest + 2D CNN	99	94.6	–
[39]	Modified 3D CNN	99.97	98.96	–
[40]	Object detection + LSTM	–	98	98.21
[41]	Object detection + 3D CNN	99.9	96	98
[42]	Two-cascade TSM	–	98.995	97.959
[43]	Dual-stream CNN + echo state network	–	99	99.01
[44]	Vision-based fight detection	100	98	–
[45]	Edge Vision	–	98.5	–
[46]	EvoKeyNet + DeepkeyFrm	–	98.98	99.29
[47]	2D CNN + ESM + STA	100	99.7	98.53
Proposed	ResNet152V2 + BiGRU	100	100	100

The experimental results obtained in the course of this study show that BiGRU performs better in terms of time than LSTM. BiGRU also achieves good performance on all data sets used in this study. The ResNet152V2-BiGRU combination achieves the best accuracy and F1 score values on all datasets.

In general, using ResNet152V2 for feature extraction improves the performance of the model on all datasets, but this increases the time required to process the test data. A difference of approximately 6 s can still be tolerated for the crowd dataset considering that the accuracy obtained increased to 0.263.

Limitations and Future Work

Despite the promising results, this study is limited by the scale and scope of the benchmark datasets used. The models were trained and tested on controlled datasets which may not fully capture the challenges of real-world surveillance, such as severe occlusions, extreme lighting variations, dynamic camera angles, as well as dense and complex crowd behavior. Consequently, the performance reported here might not directly translate to operational environments.

A primary direction for future research is to validate and retrain the proposed model on larger, more diverse, and more challenging real-world video datasets. Furthermore, exploring the model's robustness to attacks and its performance in low-resolution, long-duration video streams will be essential for practical public safety applications.

References

[1] S. Natha *et al.*, "Deep BiLSTM Attention Model for Spatial and Temporal Anomaly Detection in Video Surveillance", *Sensors*, vol. 25, art. no. 251, 2025 (https://doi.org/10.3390/s25010251).

[2] M. Karim *et al.*, "Human Action Recognition Systems: A Review of the Trends and State-of-the-art", *IEEE Access*, vol. 12, pp. 36372–36390, 2024 (https://doi.org/10.1109/access.2024.3373199).

[3] Ragedhaksha, Darshini, Shahil, and J. Arunnehru, "Deep Learning-based Real-world Object Detection and Improved Anomaly Detection for Surveillance Videos", *Materials Today: Proceedings*, vol. 80, pp. 2911–2916, 2023 (https://doi.org/10.1016/j.matpr.2021.07.064).

[4] S. Singla and R. Chadha, "Detecting Criminal Activities from CCTV by Using Object Detection and Machine Learning Algorithms", *2023 3rd International Conference on Intelligent Technologies (CONIT)*, Hubli, India, 2023 (https://doi.org/10.1109/conit59222.2023.10205699).

[5] V. Payghode *et al.*, "Object Detection and Activity Recognition in Video Surveillance Using Neural Networks", *International Journal of Web Information Systems*, vol. 19, pp. 123–138, 2023 (https://doi.org/10.1108/ijwis-01-2023-0006).

[6] P. Negre *et al.*, "Literature Review of Deep-learning-based Detection of Violence in Video", *Sensors*, vol. 24, art. no. 4016, 2024 (https://doi.org/10.3390/s24124016).

[7] T. Santos, H. Oliveira, and A. Cunha, "Systematic Review on Weapon Detection in Surveillance Footage through Deep Learning", *Computer Science Review*, vol. 51, art. no. 100612, 2024 (https://doi.org/10.1016/j.cosrev.2023.100612).

[8] J. Ruiz-Santaquiteria *et al.*, "Improving Handgun Detection through a Combination of Visual Features and Body Pose-based Data", *Pattern Recognition*, vol. 136, art. no. 109252, 2023 (https://doi.org/10.1016/j.patcog.2022.109252).

[9] L.M. Salim and Y. Celik, "Detection of Dangerous Human Behavior by Using Optical Flow and Hybrid Deep Learning", *Electronics*, vol. 13, art. no. 2116, 2024 (https://doi.org/10.3390/electronics13112116).

[10] J. Mahmoodi and A. Salajegheh, "A Classification Method Based on Optical Flow for Violence Detection", *Expert Systems with Applications*, vol. 127, pp. 121–127, 2019 (https://doi.org/10.1016/j.eswa.2019.02.032).

[11] X. Wang, J. Yang, and N. K. Kasabov, "Integrating Spatial and Temporal Information for Violent Activity Detection from Video Using Deep Spiking Neural Networks", *Sensors*, vol. 23, art. no. 4532, 2023 (https://doi.org/10.3390/s23094532).

[12] L. Hsairi, S.M. Alosaimi, and G.A. Alharaz, "Violence Detection Using Deep Learning", *Arabian Journal for Science and Engineering*, vol. 50, pp. 11669–11679, 2024 (https://doi.org/10.1007/s13369-024-09536-y).

[13] S.G. Jaiswal, S.W. Mohod, D. Sharma, and A. Hinge, "Violent Video Classification with Transfer Learning Approach Using Inception-V3 and Support Vector Machine", *Indian Journal of Science and Technology*, vol. 16, pp. 3018–3026, 2023 (https://doi.org/10.17485/ijst/v16i37.1972).

[14] M.Q. Khan, S.N. Sabir, F. Malik, and M. Khan, "Deep Convolutional Network for Automatic Violence Detection in Surveillance Videos Using Transfer Learning", *Kashf Journal of Multidisciplinary Research*, vol. 2, pp. 251–275, 2025 (https://doi.org/10.71146/kjmr270).

[15] M. Magdy, M.W. Fakhr, and F.A. Maghraby, "Violence 4D: Violence Detection in Surveillance Using 4D Convolutional Neural Networks", *IET Computer Vision*, vol. 17, pp. 282–294, 2023 (https://doi.org/10.1049/cvi2.12162).

[16] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent Flows: Real-time Detection of Violent Crowd Behavior", *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, Providence, USA, 2012 (https://doi.org/10.1109/cvprw.2012.6239348).

[17] S. Roshan, G. Srivathsan, K. Deepak, and S. Chandrakala, "Violence Detection in Automated Video Surveillance: Recent Trends and Comparative Studies", in: *The Cognitive Approach in Cloud Computing and Internet of Things Technologies for Surveillance Tracking Systems*, Academic Press, pp. 249–270, 2020 (https://doi.org/10.1016/b978-0-12-816385-6.00011-8).

[18] A. Gupta, A. Mittal, and R. Jain, "A Novel Sarcasm Detection Approach for Text-image Data: Leveraging Multimodal Fusion and Weighted Latent Factors", *Information Fusion*, vol. 103, art. no. 103266, 2025 (https://doi.org/10.1016/j.inffus.2025.103266).

[19] N. Sutranggono and R. Sarno, "Detection and Sentiment Analysis Based on Mental Disorders Aspects Using Bidirectional Gated Recurrent Unit and Semantic Similarity", *International Journal of Intelligent Engineering and Systems*, vol. 17, pp. 1–12, 2024 (https://doi.org/10.22266/ijies2024.0831.01).

[20] U. Jaishankar, J.H. Nirmal, and G. Gidaye, "Robust Time Domain Scalogram Filter Bank Feature Learning Model for Speech Depression Detection with Metaheuristic Spatio Temporal Residual BIGRU Model", *International Journal of Biomedical Engineering and Technology*, vol. 47, pp. 348–382, 2025 (https://doi.org/10.1504/ijbet.2025.145219).

[21] E.B. Nievas, O.D. Suarez, G.B. García, and R. Sukthankar, "Violence Detection in Video Using Computer Vision Techniques", *Lecture Notes in Computer Science*, vol. 6855, pp. 332–339, 2011 (https://doi.org/10.1007/978-3-642-23678-5_39).

[22] C.M. Akujuobi, "Wavelets", in: *Wavelets and Wavelet Transform Systems and Their Applications: A Digital Signal Processing Approach*, Switzerland: Springer, pp. 13–44, 2022 (https://doi.org/10.1007/978-3-030-87528-2_2).

[23] M. Shafiq and Z. Gu, "Deep Residual Learning for Image Recognition: A Survey", *Applied Science*, vol. 12, art. no. 8972, 2022 (https://doi.org/10.3390/app12188972).

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-scale Image Recognition", *arXiv*, 2014 (https://doi.org/10.48550/arXiv.1409.1556).

[25] X. Yu, Z. Yu, and S. Ramalingam, "Learning Strict Identity Mappings in Deep Residual Networks", *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, USA, 2018 (https://doi.org/10.1109/cvpr.2018.00466).

[26] R.S. Kızıltepe, J.Q. Gan, and J.J. Escobar, "A Novel Keyframe Extraction Method for Video Classification Using Deep Neural Networks", *Neural Computing and Applications*, vol. 35, pp. 24513–24524, 2023 (https://doi.org/10.1007/s00521-021-06322-x).

[27] S.H. Hendi, H.B. Taher, and K.Q. Hussein, "Automated Video Events Detection and Classification Using CNN-GRU Model", *Wasit Journal of Computer and Mathematics Science*, vol. 2, pp. 77–86, 2023 (https://doi.org/10.31185/wjcms.188).

[28] T. Zhang *et al.*, "MoWLD: A Robust Motion Image Descriptor for Violence Detection", *Multimedia Tools and Applications*, vol. 76, pp. 1419–1438, 2017 (https://doi.org/10.1007/s11042-015-3133-0).

[29] S. Keceli and A.Y. Kaya, "Violent Activity Detection with Transfer Learning Method", *Electronics Letters*, vol. 53, pp. 1047–1048, 2017 (https://doi.org/10.1049/el.2017.0970).

[30] H. Rabiee, H. Mousavi, M. Nabi, and M. Ravanbakhsh, "Detection and Localization of Crowd Behavior Using a Novel Tracklet-based Model", *International Journal of Machine Learning and Cybernetics*, vol. 9, pp. 1999–2010, 2018 (https://doi.org/10.1007/s13042-017-0682-8).

[31] A.B. Mabrouk and E. Zagrouba, "Spatio-temporal Feature Using Optical Flow Based Distribution for Violence Detection", *Pattern Recognition Letters*, vol. 92, pp. 62–67, 2017 (https://doi.org/10.1016/j.patrec.2017.04.015).

[32] P. Zhou, Q. Ding, H. Luo, and X. Hou, "Violence Detection in Surveillance Video Using Low-level Features", *PLOS.One*, vol. 13, art. no. e0203668, 2018 (https://doi.org/10.1371/journal.pone.0203668).

[33] I.P. Febin, K. Jayasree, and P.T. Joy, "Violence Detection in Videos for an Intelligent Surveillance System Using MoBSIFT and Movement Filtering Algorithm", *Pattern Analysis and Applications*, vol. 23, pp. 611–623, 2020 (https://doi.org/10.1007/s10044-019-00821-3).

[34] A.S. Keceli and A. Kaya, "Violent Activity Classification with Transferred Deep Features and 3D-CNN", *Signal, Image and Video Processing*, vol. 17, pp. 139–146, 2023 (https://doi.org/10.1007/s11760-022-02213-3).

[35] M.A. Soeleman, C. Supriyanto, and D.P. Prabowo, "An Empirical Study of CNN-LSTM on Class Imbalance Datasets for Violence Video Detection", *Proc. of the 2021 International Conference on Computer, Control, Informatics and Its Applications*, pp. 81–85, 2021 (https://doi.org/10.1145/3489088.3489126).

[36] Y. Gao *et al.*, "Violence Detection Using Oriented Violent Flows", *Image and Vision Computing*, vol. 48, pp. 37–41, 2016 (https://doi.org/10.1016/j.imavis.2016.01.006).

[37] J. Mahmoodi, H. Nezamabadi-pour, and D. Abbasi-Moghadam, "Violence Detection in Videos Using Interest Frame Extraction and 3D Convolutional Neural Network", *Multimedia Tools and Applications*, vol. 81, pp. 20945–20961, 2022 (https://doi.org/10.1007/s11042-022-12532-9).

[38] I. Serrano, O. Deniz, J.L. Espinosa-Aranda, and G. Bueno, "Fight Recognition in Video Using Hough Forests and 2D Convolutional Neural Network", *IEEE Transactions on Image Processing*, vol. 27, pp. 4787–4797, 2018 (https://doi.org/10.1109/TIP.2018.2845742).

[39] W. Song *et al.*, "A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks", *IEEE Access*, vol. 7, pp. 39172–39179, 2019 (https://doi.org/10.1109/access.2019.2906275).

[40] F.U.M. Ullah *et al.*, "An Intelligent System for Complex Violence Pattern Analysis and Detection", *International Journal of Intelligent Systems*, vol. 37, pp. 10400–10422, 2022 (https://doi.org/10.1002/int.22537).

[41] F.U.M. Ullah *et al.*, "Violence Detection Using Spatiotemporal Features with 3D Convolutional Neural Network", *Sensors*, vol. 19, art. no. 2472, 2019 (https://doi.org/10.3390/s19112472).

[42] Q. Liang, Y. Li, B. Chen, and K. Yang, "Violence Behavior Recognition of Two-cascade Temporal Shift Module with Attention Mechanism", *Journal of Electronic Imaging*, vol. 30, art. no. 043009, 2021 (https://doi.org/10.1117/1.jei.30.4.043009).

[43] W. Ullah *et al.*, "Intelligent Dual Stream CNN and Echo State Network for Anomaly Detection", *Knowledge-Based Systems*, vol. 253, art. no. 109456, 2022 (https://doi.org/10.1016/j.knosys.2022.109456).

[44] Ş. Aktı, G.A. Tataroğlu, and H.K. Ekenel, "Vision-based Fight Detection from Surveillance Cameras", *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Istanbul, Turkey, 2019 (https://doi.org/10.1109/IPTA.2019.8936070).

[45] F.U.M. Ullah *et al.*, "AI-assisted edge vision for violence detection in IoT-based industrial surveillance networks", *IEEE Transactions on Industrial Informatics*, vol. 18, pp. 5359–5370, 2021 (https://doi.org/10.1109/TII.2021.3116377).

[46] M. Shoaib *et al.*, "Augmenting the Robustness and Efficiency of Violence Detection Systems for Surveillance and Non-surveillance Scenarios", *IEEE Access*, vol. 11, pp. 123295–123313, 2023 (https://doi.org/10.1109/access.2023.3329062).

[47] J. Mahmoodi and H. Nezamabadi-pour, "A Spatio-temporal Model for Violence Detection Based on Spatial and Temporal Attention Modules and 2D CNNs", *Pattern Analysis and Applications*, vol. 27, art. no. 46, 2024 (https://doi.org/10.1007/s10044-024-01265-0).

Khaled Merit, Ph.D.
Laboratory of TIT, Department of Electrical Engineering
https://orcid.org/0000-0002-7762-1898
E-mail: merit.khaled@univ-bechar.dz
Tahri Mohammed University of Bechar, Algeria
https://www.univ-bechar.dz

Mohammed Beladgham, Ph.D., Full Professor
Laboratory of TIT, Department of Electrical Engineering
https://orcid.org/0000-0002-2371-6859
E-mail: beladgham.mohammed@univ-bechar.dz
Tahri Mohammed University of Bechar, Algeria
https://www.univ-bechar.dz

Lightweight Flow-based Anomaly Detection for IoT Using HC-MTDNN: A Hierarchically Cascaded Multitask Deep Neural Network

Mohamed Amine Beghoura and Younes Belouche

University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria

https://doi.org/10.26636/jtit.2025.4.2311

Abstract — In this article, we propose a lightweight, hierarchical multi-task learning framework designed for detecting both high-level and fine-grained threats in IoT traffic. The developed model focuses on anomalies detectable through flow-level metadata. The deliberate choice to prioritize computational efficiency by excluding content analysis scopes the approach to payload-independent threats, while still enabling robust detection of key attack classes. To further enhance efficiency within this metadata-driven paradigm, we introduce HC-MTDNN, a hierarchical multitask model that integrates a gated feature mechanism and feature reuse to significantly reduce redundancy and computational overhead, improving upon previous hierarchical architectures and achieving high performance while dealing with volumetric and protocol-based attacks. The model is evaluated on four benchmark datasets: CICIoT2023, N-BaIoT, Bot-IoT, and Edge-IIoTset. It demonstrates strong performance in both binary and multiclass classification tasks, with an average inference time of 122 µs per sample and a compact model size of 2.4 MB. The proposed framework effectively balances accuracy and computational efficiency, offering a practical and scalable solution for securing resource-constrained IoT environments.

Keywords — anomaly detection, deep neural network, IoT security, lightweight model, multitask learning, network traffic analysis

1. Introduction

The exponential growth of the Internet of Things (IoT) has transformed conventional networks into vast interconnected ecosystems spanning various domains such as smart homes, healthcare, industrial automation, and smart cities [1]. By 2025, more than 75 billion IoT devices are projected to be online, roughly 40% of them in smart home environments [2]. Although this expansion offers convenience and functionality, it also increases vulnerability to attacks. Many IoT devices operate under severe resource limitations and lack robust security mechanisms, making them attractive targets for cyber threats [3], [4]. In particular, high-profile attacks such as the Mirai botnet [5] have demonstrated the dangers associated with compromised IoT infrastructures.

IoT anomaly detection is particularly sophisticated due to the heterogeneity, volume, and nonstationarity of the data generated from such applications. Traditional detection methods, such as rule-based systems and fixed thresholds, cannot cope with the dynamic and multimodal characteristic of traffic in IoT networks [6]. As a result, these methods frequently misclassify benign traffic as malicious, leading to an excessive number of false alarms.

However, most of the existing approaches in IoT anomaly detection handle each classification task independently, employing separate single-task models for binary classification, attack-type categorization, and fine-grained subtype classification. The authors of [7], [8] each deploy distinct single-task models dedicated specifically to binary or multiclass scenarios without exploiting task interdependencies. Such isolated training can neglect beneficial shared representations across tasks, potentially limiting generalization and computational efficiency.

To address these challenges, we propose HC-MTDNN, a hierarchical multitask model designed to tackle three core IoT anomaly detection tasks in a single forward pass. Instead of treating each task independently, the model cascades predictions, refining the output at each level. HC-MTDNN performs anomaly detection in a staged, multitask manner, progressing from binary to coarse to fine classification. Each level of the network is responsible for different interdependent tasks, such as feature extraction, anomaly detection, and classification. By cascading these tasks, the model can progressively refine its analysis, leading to a more accurate identification of anomalies. This structure allows the model to efficiently process data and reduce false positives.

Incorporation of multitask learning (MTL) [9] enables the HC-MTDNN to simultaneously learn and optimize multiple related tasks. This approach leverages shared representations, improving generalization between tasks, and enhancing the model's ability to detect various types of anomalies.

The primary advantage of the proposed hierarchical multitask architecture lies in its ability to progressively refine classification through shared supervision and structured information flow. HC-MTDNN processes each IoT flow in three gated stages:

- benign vs. malicious screening,
- attack-family categorization,

- fine-grained subtype classification.

Intermediate logits from one task are concatenated or transformed before being passed to the next stage, allowing downstream classifiers to condition on earlier decisions. The shared encoder allows for generalization and reduces redundant computation.

Additionally, the architecture strengthens feature reuse, stabilizes rare subcategories, and diversifies error signals. The model also maintains hierarchical consistency by using gating and attention mechanisms to align predictions across levels, thus minimizing contradictions like predicting "mirai.scan" when the binary classifier identifies a sample as benign.

Finally, this multitask setup often leads to faster convergence and improved generalization, as the shared encoder can absorb additional tasks with minimal reconfiguration, making the model adaptable to evolving threat landscapes in complex IoT environments.

The proposed model is designed for flow-based anomaly detection, using network metadata such as packet counts, flow durations, and protocol attributes to identify threats. This approach excels at detecting volumetric attacks (e.g., DDoS) and protocol anomalies, but is inherently limited for payload-intensive threats (e.g., SQL injections or XSS), where content analysis is required. This trade-off ensures deployability on edge devices, prioritizing speed and low resource usage over comprehensive payload inspection.

With 245 249 total parameters, the model occupies just 2.4 MB of disk space. The average inference latency is 122 µs per sample, corresponding to roughly 8 200 predictions per second, which falls within the real-time requirements for gateway-level traffic inspection. This parameter sharing and conditional gating eliminate redundant computation, allowing HC-MTDNN to be deployable on memory and power constrained IoT edge devices.

We evaluate the proposed model across four datasets: CICIoT2023 [10], N-BaIoT [11], Bot-IoT [12], and EdgeIIoT [13]. These datasets cover diverse environments and traffic profiles. They span distinct deployment scenarios: CICIoT2023 imitates modern smarthome traffic, N-BaIoT isolates single device compromises typical of consumer gadgets, Bot-IoT replicates campus-scale probing and DDoS, while EdgeIIoT captures latency-sensitive industrial control flows.

Across all experiments, HC-MTDNN delivers accuracy and efficiency that match the practical constraints of edge hardware. On the binary task, it attains macro F1 between 97% and 100% on every dataset, while at the coarse-level it reaches 99.5% accuracy on CICIoT2023, 97% on EdgeIIoT and Bot-IoT, and a near-perfect 99.99% on N-BaIoT. Even at the most demanding fine-grained level (34 classes in CICIoT2023, 9 – 10 classes on the other sets) the network keeps the weighted F1 above 93% for Bot-IoT and the macro F1 above 83% for the heavily imbalanced N-BaIoT. The results demonstrate robust multitask performance across all classification hierarchies, highlighting the robustness and effectiveness in accurately detecting and classifying anomalies within complex IoT environments.

The remainder of this paper is structured as follows. Section 2 reviews related work and discusses the limitations of existing approaches. Section 3 presents, in detail, the proposed architecture of the HC-MTDNN model. Section 4 describes the datasets, experimental setup, and evaluation metrics used. Section 5 reports and analyzes the experimental results, comparing them with established baselines. Finally, Section 6 summarizes the findings, implications, and suggests directions for future research.

2. Related Work

The extensive deployment of IoT devices across various sectors has significantly increased the prevalence of cyber attacks, underscoring the need for effective anomaly and intrusion detection mechanisms [14]. Traditional anomaly detection strategies, such as rule-based systems and fixed threshold methods, frequently encounter difficulties due to the heterogeneous, high-volume, and dynamically changing nature of IoT-generated data. Such conventional methods often produce false alarms or fail to identify subtle attacks, particularly in environments where multiple distinct data streams are processed simultaneously. A summary of recent intrusion detection studies is provided in Tab. 1 which categorizes the approaches by model type, dataset, key contributions, and gaps addressed by the proposed method.

2.1. Traditional Machine Learning Approaches

Several recent studies have applied traditional machine learning (ML) approaches using the CICIoT2023 dataset [10]. The authors of [8] introduced a random forest-based intrusion detection framework specifically addressing class imbalance, achieving notable performance improvements of 3.72% in precision, 3.75% in recall and 4.69% in F1 score compared to existing methodologies. Their method also showed an enhancement of 7.9% in the F1 score for underperforming classes. In another comparative analysis, multiple machine learning algorithms, including logistic regression, AdaBoost, perceptron, MLP, random forest (RF) and hist-gradient boosting, were evaluated for different classification scenarios (binary, eight class and 34-class), with RF outperforming others in accuracy, while hist-gradient boosting excelled in computational efficiency [15].

Addressing specific attack types, the researchers developed a specialized intrusion detection system (IDS) that employs hierarchical feature selection coupled with the CatBoost algorithm, targeting DoS, DDoS, and Mirai attack variants. This approach achieved fast prediction times and high accuracy, significantly improving cybersecurity defenses against sophisticated threats [16]. Similarly, a comprehensive evaluation, as presented in [7], emphasized the broad spectrum of threats encapsulated by the CICIoT2023 dataset, reinforcing its utility in benchmarking classification methods. Parallel, lightweight ML models – such as decision trees, closest neighbors k, RF,

and naive Bayes – were evaluated, demonstrating impressive precision and processing efficiency, notably the ability of decision trees to classify nearly three million instances per second [17].

In [18], refined preparation and feature selection phases are investigated through cooperative game theory, and RF achieves 99% accuracy on the original CICIoT2023 dataset. However, the accuracy decreased slightly with novel features, highlighting complexities in feature engineering for IoT intrusion detection. While these single-task machine learning approaches achieve high accuracy on balanced classes and specific attack types, they often treat classification tasks independently, overlooking intertask dependencies such as shared patterns between binary detection and multiclass categorization. This leads to redundant computations and limited generalization of diverse or evolving threats.

2.2. Multitask Learning and Lightweight Models

To overcome limitations inherent in single-task or parallel-output-head models, researchers have explored multitask learning frameworks, aiming to enhance anomaly detection performance by leveraging interrelated tasks [19]. The CICIoT2023 dataset, a comprehensive and realistic benchmark, has been widely utilized to evaluate the effectiveness of these advanced models, particularly emphasizing improvements in the detection of low-profile attacks [20]. Additionally, due to the resource-constrained nature of many IoT devices, significant research has focused on developing lightweight models optimized for efficient deployment in such environments.

In resource-sensitive IoT scenarios, the authors of [21] introduced DL-BiLSTM, integrating DNN and bi-LSTM networks, along with incremental PCA and dynamic quantization to optimize model performance for limited-resource environments. Furthermore, in [22], an innovative VGGIncepNet model was proposed that converts non-image network data into image format to leverage CNN feature extraction capabilities, significantly outperforming established NLP-based models such as BERT and XLNet in CICIoT2023.

In [23], edge-based deep learning models are presented that employ 1D-CNN architectures optimized by preprocessing techniques that address data imbalance and distribution discrepancies, achieving a robust F1 score of 93.8%. Furthermore, the authors of [24] proposed a cost-sensitive autoencoder (CSAE)-based ensemble approach, demonstrating exceptional accuracy rates for both binary and multiclass classifications.

In article [25], a DGConv-IDS was developed. It is a lightweight autoencoder and CNN-based model tailored for resource-limited IoT environments. The model used sliding-window techniques to manage computational overhead while providing real-time DDoS detection. Similarly, in [26], the convolutional Kolmogorov-Arnold network (CKAN) is introduced which integrates Kolmogorov-Arnold layers into convolutional neural networks, achieving high performance with fewer parameters. The authors of [27] proposed hybrid models, such as the autoencoder-CNN and transformer-DNN frameworks, emphasizing reshaping network traffic, handling class imbalance, and improving feature extraction capabilities across multiple datasets, including CICIoT2023.

Existing lightweight deep learning models prioritize computational efficiency and edge deployment, but often lack hierarchical structures for progressive refinement from binary to fine-grained classification, leading to potential error propagation in multiclass scenarios. Moreover, they often incorporate conditional mechanisms such as dynamic gating to adapt features based on prior task outputs.

2.3. Dataset-specific and Hybrid Models

The authors of [28] propose an efficient anomaly detection mechanism for IoT architectures using DNN, with a specific focus on feature selection through mutual information (MI). The study uses the Bot-IoT 2020 dataset and evaluates the performance of several deep learning models, including DNN, CNN, RNN, and RNN variants. The authors demonstrate that selecting the top 16 to 35 MI-based features, instead of using all 80 features, resulted in only negligible performance degradation while significantly reducing model complexity. The proposed DNN-based model achieves an accuracy of 99.01% with a false alarm rate (FAR) of 3.9%.

In [29], XAI-IoT, an explainable AI framework is introduced designed to enhance multi-class anomaly detection and defect type classification in IoT systems. The framework incorporates seven explainable AI (XAI) techniques, including SHAP, LIME, CEM, and LOCO, to evaluate the importance in model predictions. Experimental validation was performed on two datasets: one collected from IoT-based MEMS sensors and the other from IoT botnet attacks (N-BaIoT). The results indicate that single-model approaches delivered better performance on the MEMS dataset, while ensemble-based models outperformed on the N-BaIoT dataset. The use of XAI techniques allowed the identification of critical features that influenced model decisions in both contexts.

In [30], an IDS for detecting DoS attacks in IoT networks by relying on ML algorithms is described. The study compared four classifiers: decision tree (DT), RF, K-nearest neighbor (kNN), and support vector machine (SVM), to determine the most effective model for classifying DoS traffic. Feature selection was enhanced using correlation-based feature selection (CFS) and a genetic algorithm (GA), with the IoTID20 data set used for training, which includes real-time traffic data with simulated DoS attacks. The DT and RF classifiers, using GA-selected features (13 features), achieved 100% accuracy, precision, recall, and F1 scores. The DT model outperformed RF in terms of computational efficiency. The study emphasizes the effectiveness of the IoTID20 dataset and the chosen feature selection methods to improve the performance of the model.

The authors of [31] introduce DeepDetect, a hybrid deep learning model for anomaly detection in IoT networks which combines CNN, GRU, and Bi-LSTM to improve network traffic analysis. The hierarchical CNN structure captures spatial features, while the problem of GRU mitigates the vanishing

Tab. 1. Summary of recent intrusion detection studies in IoT environments.

Ref.	Model type	Dataset	Key contribution
[8]	RF	CICIoT2023	Improved class performance; tackled class imbalance
[7]	ML	CICIoT2023	Comprehensive benchmarking across attack categories
[18]	RF + game theory	CICIoT2023	Cooperative game theory feature selection
[21]	DL-BiLSTM+PCA	CICIoT2023	Resource-efficient BiLSTM with dynamic quantization
[22]	VGGIncepNet (CNN-based)	CICIoT2023	Converted traffic to images; outperformed BERT/XLNet
[23]	1D-CNN	CICIoT2023	Edge-based detection with preprocessing for imbalance
[24]	CSAE	CICIoT2023	High accuracy for binary and multiclass tasks
[25]	DGConv-IDS	CICIoT2023	Real-time DDoS detection via sliding windows
[26]	CKAN	CICIoT2023	Low-parameter model with high performance
[27]	AE-CNN, transformer-DNN	CICIoT2023	Multi-dataset approach; class imbalance handling
[28]	DNN, CNN, RNN variants	BoT-IoT 2020	Mutual-information feature selection (top 16 – 35 features)
[29]	Ensemble + SHAP, LIME	N-BaIoT, MEMS	XAI-IoT with comparative model/XAI analysis
[30]	DT, RF, kNN, SVM	IoTID20	GA-based feature selection; 100% metrics with DT
[31]	CNN-GRU-BiLSTM	NSL-KDD	High accuracy; temporal modeling with low FAR
[32]	XGBoost, RF	IoT-23 combined	PySpark-based scalable real-time IDS
[33]	TinyML + DT, RF, KNN	Custom + real IoT LAN data	Energy/memory efficient IDS using TinyML
[34]	CNN + LSTM/-GRU/BiLSTM	NSL-KDD, BoT-IoT, MQTTset	Addressed class imbalance with SMOTE and class weighting
[35]	RF vs. DNN	CICIoT2023	Multilevel classification and feature selection study
[36]	Transformer, CNN + LSTM, DNN	CICIoT2023	Multi-class top accuracy with transformer
[37]	PCA + expansion – compression NN	UNSW-NB15, Bot-IoT	Lightweight NN with NID loss; 99.99% binary accuracy
[38]	Multi-stage pipeline	CIC-IDS-2017, CSE-CIC-IDS-2018	Adjustable zero-day detection; low bandwidth/latency
[39]	MI + attention CNN	Edge-IoTset, IoTID20, ToN IoT, CIC-IDS2017	99.81% average accuracy; attention helps low-instance classes
[40]	Multitask LSTM + feature selection	IoT-23, EU CEF VAR-IoT, 18-device pcaps	Joint malware detection/identification; SMOTE-ENN + XGBoost-/SULOV

gradients and learns sequential dependencies. The Bi-LSTM captures long-term dependencies from both forward and backward contexts, improving temporal analysis. Based on the NSL-KDD dataset, DeepDetect achieved 99.12% accuracy for binary classification and 99.31% for multiclass classification, demonstrating superior performance with a lower false positive rate and higher detection rate compared to other deep learning-based IDS.

Paper [32] presents a real-time IDS for IoT networks using multiclass ML techniques. Using the IoT-23 combined dataset, which includes more than 1.4 million records of various types and benign traffic, the class imbalance with SMOTE and the applied SelectKBest is addressed. IDS was built on a PySpark architecture to support scalable training and inference. Five ML models were tested using a one-versus-rest approach, with XGBoost achieving the highest accuracy (98.89%) and RF delivering the fastest inference time (0.0311 s), demonstrating a strong balance between speed and accuracy.

In [33], an ML-based IDS is developed tailored for resource-constrained IoT devices, emphasizing energy and memory efficiency. By integrating TinyML with traditional ML models, the study addressed key challenges in limited resource environments. A major contribution was the creation of a rich validation dataset combining prior work, laboratory experiments, and real-world metrics across IoT layers (end devices, edge, cloud), including both normal and malicious traffic. The system was tested in a LAN based setup with extended edge/cloud components, using models such as DT

(99.5% accuracy), KNN (96.5%), Naive Bayes (97.3%) and RF (98.3%). The results highlighted a trade-off between accuracy and training time, with more accurate models requiring longer training.

The authors of [34] propose a DL-based anomaly detection framework for IoT networks, utilizing a combination of RNN and CNN. Their study developed lightweight models that employ LSTM, BiLSTM, and GRU architectures to perform binary and multiclass classification tasks. CNNs were also incorporated for feature selection to enhance detection performance. Models were trained and evaluated on several widely used datasets, including NSL-KDD, BoT-IoT, IoT-NI, IoT-23, MQTT, MQTTset, and IoT-DS2. To address the issue of class imbalance within these datasets, the authors applied class weighting techniques during training and employed the Borderline-SMOTE algorithm to generate synthetic samples and balance the training data distribution.

In [35], an anomaly detection study is conducted in IoT-based healthcare systems using the CICIoT2023 data set. Their investigation involved multilevel classification architectures, including 2-class (binary), 8-class, and 34-class models. The authors explored two training approaches: one using the full set of features and the other using a reduced feature subset. To address class imbalance, they applied SMOTE. Their evaluation demonstrated that, on both training paths and on the balanced CICIoT2023 dataset, the RF classifier consistently outperformed the DNN model.

These dataset-specific and hybrid models demonstrate strong performance on individual benchmarks but typically do not integrate multitask hierarchies for handling interdependent classification tasks, such as simultaneous binary and fine-grained detection. This results in missed opportunities for shared learning and efficiency in resource-constrained settings.

2.4. Advanced Hybrid and Attention-based Models

The authors of [36] propose a transformer-based IDS evaluated on the CICIoT2023 dataset. Their model leveraged self-attention mechanisms to effectively handle multi-class intrusion detection, achieving a high accuracy of 99.40%. After comparing seven neural network models, they found the transformer to be the most effective solution for multiclass tasks, while DNN and CNN+LSTM models performed best for binary classification.

In [37], a lightweight neural network-based IDS is presented that uses PCA for feature dimensionality reduction. It relies on an expansion compression classifier architecture with inverse residual blocks and channel shuffle operations to minimize computational cost, and a loss of NID to mitigate class imbalance. Evaluated on UNSW-NB15 and Bot-IoT, it achieves a precision of up to 99.99% (F1 98.81%) for binary detection and multiclass accuracies of 86.11% and 96.15%, respectively, without altering its core architecture.

Work [38] introduces a multi-stage approach for hierarchical IDS with a three-stage anomaly detection pipeline: fast filtering by anomaly score, confidence-based attack classification, and strict thresholding to separate unknown attacks from false positives, enabling efficient, adjustable detection of binary and multiclass intrusions, including zero-day attacks, in the CIC-IDS2017 and CSE-CIC-IDS-2018 datasets. Each stage can be deployed independently to minimize bandwidth usage and prediction latency without requiring retraining.

The authors of [39] propose an attention-based CNN for IDS, using MI for feature selection and an attention mechanism to improve learning in low-instance classes. The proposal is evaluated in Edge-IoTset, IoTID20, ToN IoT, and CIC-IDS2017. It achieves an average accuracy of 99.81%, with 98.02% precision, 98.18% recall, and an F1 score of 98.08%.

In [40], a multitask LSTM is proposed for the detection and identification of IoT malware through behavioral traffic analysis, performing both benign/malicious classification and malware type prediction on 145 pcaps from 18 devices and on the IoT-23 and EU CEF VARIoT datasets. Features are organized into flow-, flag-, and payload-related modalities, each subjected to recursive XGBoost and SULOV feature selection before merging, with class imbalance addressed using SMOTE-ENN and extensive experiments on imbalance techniques, feature selection, and modality fusion.

Advanced hybrid and attention-based models enhance temporal and spatial feature analysis, but often miss conditional feature modulation, where prior task outputs dynamically influence subsequent processing. This can limit adaptability in hierarchical scenarios.

Although significant progress has been made in IoT anomaly detection, existing approaches typically address tasks individually, deploying separate single-task models, thus overlooking valuable shared information among related classification tasks. Moreover, hierarchical cascading, progressively refining anomaly classification from broad to detailed levels, has rarely been explicitly combined with computational efficiency strategies optimized for deployment in resource-constrained IoT environments. Most lightweight models neglect hierarchical classification or fail to incorporate dynamic feature gating and hierarchical feature reuse mechanisms necessary for efficiency in real-time scenarios.

Table 1 summarizes mentioned studies on intrusion detection.

3. Proposed Method

3.1. Model Architecture

The proposed architecture, as illustrated in Fig. 1, is a multi-stage neural network processing incoming IoT sensor streams using a shared feature encoder that learns a common representation. The data are then refined by three task-specific heads: a binary anomaly detector, a coarse (categorical) classifier, and a fine-grained (subcategorical) classifier.

This hierarchical structure enables the model to progressively refine the predictions: first, by identifying anomalies, then by categorizing broad attack types, and finally by distinguish specific subcategories.

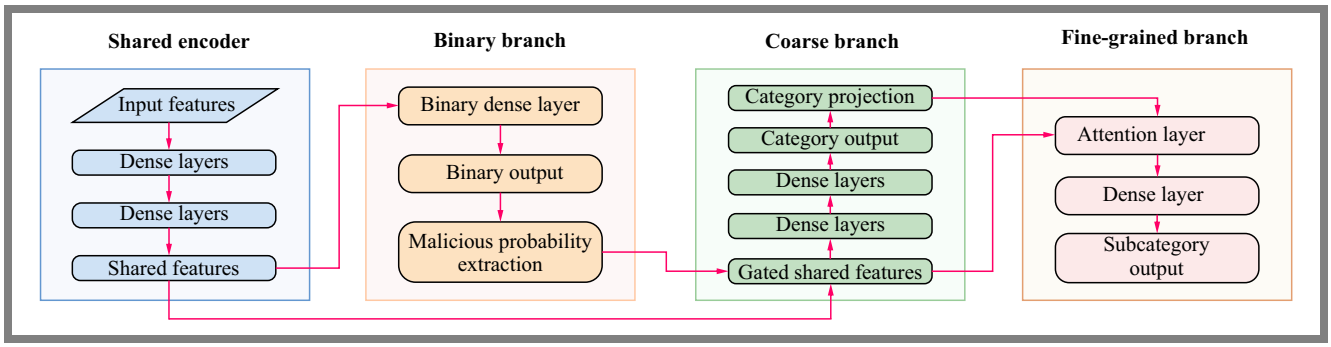

Fig. 1. Overall architecture of the proposed HC-MTDNN model.

The core of the architecture is a deep neural backbone that extracts general-purpose features from raw IoT data. This encoder is hard-shared across all tasks, ensuring that early layers capture patterns common to anomaly detection, while later layers refine these features for task-specific objectives. Shared encoders are a standard approach in multi-task learning to reduce redundancy and improve generalization.

The binary branch is the lightweight head that receives the shared features and outputs a scalar anomaly probability indicating whether the input is normal or anomalous. The output is also reused as a gating signal to modulate downstream processing. Gating mechanisms are commonly used in attention-based architectures to dynamically suppress irrelevant features.

The coarse branch performs coarse-grained anomaly classification. It applies an attention mechanism to the shared features, refining them using the binary gating signal and preliminary category logits. The attention mechanism computes feature-wise importance weights, focusing on dimensions most relevant to distinguishing broad anomaly categories.

The fine-grained branch identifies fine-grained anomaly subtypes. It takes as input a fusion of attention-refined features from the coarse branch, the binary gating signal, and the predicted category. This hierarchical design mirrors strategies used in multilevel classification tasks, where coarse predictions inform finer distinctions.

All components are jointly trained end-to-end. The shared encoder is updated by all three tasks, encouraging it to learn useful features in binary, categorical, and subcategorical decisions. Selective feature flow ensures that computational effort is focused where needed the most (e.g., suppressing processing for benign samples).

3.2. Input Representation and Shared Encoder

The model processes a static feature vector $\mathbf{x} \in \mathbb{R}^d$, which encapsulates critical IoT traffic characteristics.

These include network flow statistics (e.g., packet counts, byte rates), packet-level attributes (i.e., protocol types, payload sizes), and temporal dynamics such as traffic variations over sliding windows. The shared encoder consists of three dense layers with ReLU activation (256, 256, and 128 neurons, respectively), designed to extract foundational representations while minimizing redundancy between tasks. The final output of the shared encoder is defined as:

$$\mathbf{x}_{\text{shared}} = \text{ReLU}\left[\text{Dense}_{128}\left(\text{ReLU}(\text{Dense}_{256}(\mathbf{x}))\right)\right], \quad (1)$$

where nested ReLU activations ensure non-linearity at each layer.

This design aligns with multitask learning principles, enabling knowledge transfer by learning task-agnostic representations.

3.3. Binary Classification Branch

The binary classification branch employs a shallow structure comprising a single dense layer (128 neurons, ReLU activation) followed by a softmax output:

$$\hat{\mathbf{y}}_{\text{bin}} = \text{softmax}\left[\text{Dense}_2\left(\text{Dense}_{128}(\mathbf{x}_{\text{shared}})\right)\right]. \quad (2)$$

This prioritizes computational efficiency for edge deployment, balancing accuracy and inference speed. The output $\hat{\mathbf{y}}_{\text{bin}}$ serves dual purposes: direct binary anomaly detection (normal vs. anomalous) and generating a gating signal $p_{\text{malicious}}$ to modulate downstream processing.

3.4. Gated Coarse Classification Branch

To refine predictions, the malicious probability $p_{\text{malicious}}$ is extracted from $\hat{\mathbf{y}}_{\text{bin}}$ via a lambda layer:

$$p_{\text{malicious}} = \Lambda\left(z \mapsto z[:,1]\right)(\hat{\mathbf{y}}_{\text{bin}}), \quad (3)$$

where $z[:,1]$ isolates the probability of the anomalous class.

A sigmoid-activated dense layer then generates gating weights $\mathbf{g} \in [0,1]^{128}$:

$$\mathbf{g} = \sigma\left(\text{Dense}_{128}(p_{\text{malicious}})\right). \quad (4)$$

These gating weights dynamically modulate shared features by selectively emphasizing relevant dimensions and suppressing irrelevant ones, particularly for benign samples. Formally, this modulation is implemented as element-wise multiplication between gating weights and shared features.

$$\mathbf{x}_{\text{gated}} = \mathbf{x}_{\text{shared}} \odot \mathbf{g}. \quad (5)$$

Dynamic gating significantly reduces unnecessary computations by minimizing redundant feature processing for benign inputs, enhancing computational efficiency crucial for resource-constrained IoT environments. The gated features are then processed by two dense layers (128 neurons, ReLU)

to produce coarse-grained classification outputs:

$$\hat{\mathbf{y}}_{\text{coarse}} = \text{softmax}\left[\text{Dense}_{\text{coarse}}\left(\text{Dense}_{128}(\mathbf{x}_{\text{gated}})\right)\right] . \quad (6)$$

3.5. Fine-grained Classification Branch

The fine-grained classification branch incorporates two components:

Semantic projection is the 8-class output, where $\hat{\mathbf{y}}_{\text{coarse}}$ is projected into the shared feature space to embed coarse-grained priors:

$$\mathbf{x}_{\text{proj}} = \text{ReLU}\left(\text{Dense}_{256}(\hat{\mathbf{y}}_{\text{coarse}})\right) . \quad (7)$$

This projection aligns the semantic context with latent features, enhancing cross-task knowledge transfer.

The **cross-task attention** mechanism fuses $\mathbf{x}_{\text{shared}}$ and $\mathbf{x}_{\text{proj}}$:

$$\mathbf{x}_{\text{att}} = \text{Attention}\left(\mathbf{x}_{\text{shared}}, \mathbf{x}_{\text{proj}}, \mathbf{x}_{\text{proj}}\right) , \quad (8)$$

where queries $\mathbf{x}_{\text{shared}}$ and keys/values $\mathbf{x}_{\text{proj}}$ compute feature-wise importance weights.

The attended vector $\mathbf{x}_{\text{att}}$ is concatenated with $\mathbf{x}_{\text{gated}}$:

$$\mathbf{x}_{\text{concat}} = \text{Concat}(\mathbf{x}_{\text{gated}}, \mathbf{x}_{\text{att}}) . \quad (9)$$

This combined representation is passed through a dense layer (ReLU) and batch normalization:

$$\mathbf{x}_{\text{norm}} = \text{BatchNorm}\left[\text{ReLU}\left(\text{Dense}_{128}(\mathbf{x}_{\text{concat}})\right)\right] , \quad (10)$$

before yielding the final fine-grained prediction:

$$\hat{\mathbf{y}}_{\text{fine}} = \text{softmax}\left[\text{Dense}_{\text{fine}}(\mathbf{x}_{\text{norm}})\right] . \quad (11)$$

This hierarchical fusion takes advantage of coarse-level context to constrain fine-grained predictions, improving robustness for closely related subtypes.

4. Experimental Setup and Evaluation

4.1. Dataset Overview

CICIoT2023 [10] is a comprehensive benchmark data set that captures network traffic from 105 real IoT devices in a laboratory environment. It includes 33 distinct attacks across seven categories (DDoS, DoS, Reconnaissance, Web-based attacks, BruteForce, Spoofing, Mirai botnet) and benign traffic (e.g., video streaming, sensor data). Features such as flow duration, packet length, and protocol types are extracted from pcap files and stored in CSV format. Baseline models (LR, RF) have been evaluated on binary, 8-class, and 34-class tasks, making it ideal for comparative studies.

Bot-IoT [12] combines real and simulated IoT traffic with various cyberattack scenarios. Developed using a realistic testbed, it addresses limitations of older datasets (e.g., outdated attack patterns, poor labeling). Its validity has been confirmed through statistical analysis and ML experiments.

N-BaIoT [11] focuses on botnet detection, containing traffic from nine commercial IoT devices infected with Mirai and Bashlite malware. It includes over 7 million records with 115 features, classified into ten categories (primarily DDoS and remote access attacks).

Edge-IIoTset [13] is a cybersecurity data set for IoT/IIoT applications, supporting centralized and federated learning. Generated using a custom testbed, it includes 14 attack types across five categories (DoS/DDoS, information gathering, MITM, injection, malware). Of 1 176 initial features, 61 were selected based on correlation and domain knowledge, ensuring efficient model training.

4.2. Evaluation Metrics

The performance of the hierarchical multitask DNN is assessed using accuracy, precision, recall, F1 score, and AUC-ROC. The definitions of the aforementioned terms are provided below:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} , \quad (12)$$

$$\text{Precision} = \frac{TP}{TP + FP} , \quad (13)$$

$$\text{Recall} = \frac{TP}{TP + FN} , \quad (14)$$

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} , \quad (15)$$

where TP, TN, FP, and FN denote true/false positives/negatives.

The F1 score is prioritized due to class imbalance in IoT security datasets.

4.3. Model Training

The model is trained on 80% of the data (64% training, 16% validation), with 20% held for testing. Pre-processing includes label mapping and feature normalization. Hyperparameters are tuned iteratively: learning rate 10^{-3} (Adam optimizer), batch size 512, epochs 127 (one CSV file per epoch to manage memory).

The loss function is:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{bin}}\,\mathcal{L}_{\text{bin}} + \lambda_{\text{int}}\,\mathcal{L}_{\text{int}} + \lambda_{\text{fine}}\,\mathcal{L}_{\text{fine}} , \quad (16)$$

with λ_{bin}, λ_{int}, and λ_{fine} as task-specific weights.

Figure 2 shows training/validation curves. The binary head converges faster than multiclass heads, reflecting its simplicity. The validation accuracy plateaus earlier for coarse tasks, suggesting diminishing returns beyond 80 epochs.

5. Experimental Results

The proposed lightweight multitask DNN demonstrates robust performance across the IoT datasets used. With 245 249 parameters (2.4 MB in size) and an average inference time of 122 µs per flow, the model is optimized for real-time deployment on resource-constrained devices. Hierarchical classification tasks are evaluated, with macro and weighted average metrics summarized in Tab. 2.

5.1. Binary Classification

The binary classification results (Tab. 3, Fig. 3) show near-perfect or perfect separation between benign and malicious

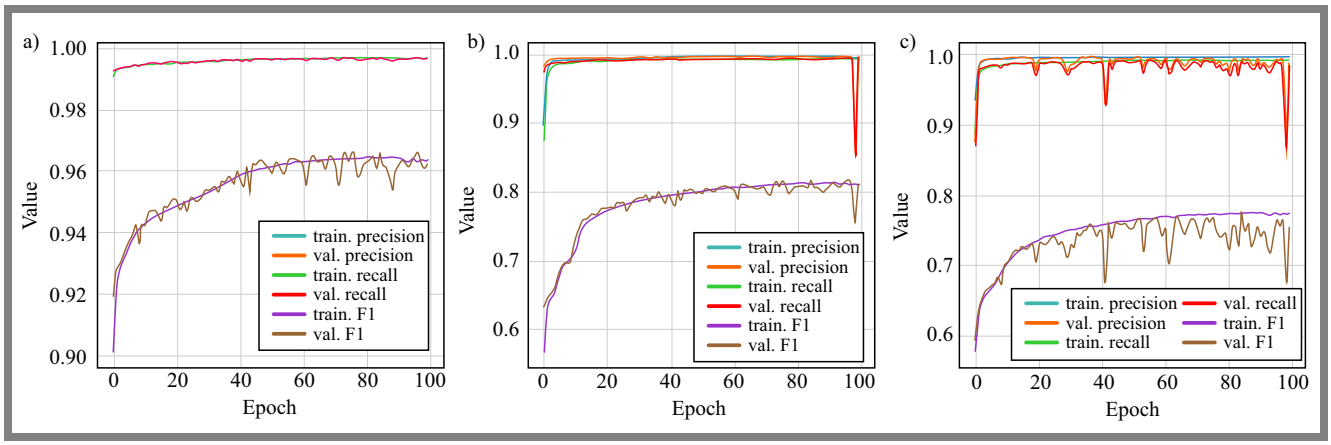

Fig. 2. Performance of training and validation of the proposed model on the CICIoT2023 data set: a) binary classification b) 8-class category classification, and c) 34-class fine-grained subtype classification.

Tab. 2. Summary of the macro- and weighted-average metrics results on the test splits of the four datasets.

Dataset	Level	Macro			Weighted		
		P	R	F_1	P	R	F_1
CICIoT 2023	Binary	96.0	97.0	97.0	100.0	100.0	100.0
	Category	92.0	78.0	82.0	99.0	99.0	99.0
	Subtype	83.0	75.6	77.2	99.2	99.2	99.2
N-BaIoT	Binary	100.0	100.0	100.0	100.0	100.0	100.0
	Category	100.0	100.0	100.0	100.0	100.0	100.0
	Subtype	97.0	89.0	87.0	91.0	88.0	83.0
Bot-IoT	Binary	98.5	96.7	97.6	99.3	98.4	98.8
	Category	99.2	95.1	97.0	99.2	97.1	98.1
	Subtype	90.4	87.9	87.7	95.0	93.7	93.6
EdgeIIoT	Binary	100.0	100.0	100.0	100.0	100.0	100.0
	Category	89.0	87.0	87.0	98.0	97.0	97.0

traffic across all data sets. N-BaIoT, Bot-IoT, and Edge-IIoT report 100% precision, recall, and F1 score, confirmed by diagonal dominance in confusion matrices. For CICIoT2023, 4% of benign flows are misclassified as malicious. This occurs because low-level attacks (i.e., reconnaissance scans) mimic benign behaviors, creating subtle overlaps in header-level features like packet size distributions and interarrival times. These patterns suggest that the model struggles to distinguish benign traffic from low-intensity adversarial activities that exploit normal protocol behaviors, such as slow port scans or HTTP GET requests.

5.2. Coarse-grained Classification

Coarse-grained classification (Tab. 4 and Fig. 4) identifies broader attack families (e.g., DDoS, DoS, Mirai, Reconnaissance). On CICIoT2023, the model attains 99.5% weighted accuracy, but struggles with underrepresented classes like web-based (40% recall) and BruteForce (24% recall). These errors arise from feature overlap in HTTP methods and port usage, where web-based attacks share characteristics with reconnaissance activities (e.g., POST requests, standard ports 80/443). Edge IIoT achieves 97.13% accuracy, though password attacks and SQL injections exhibit sub-80% precision.

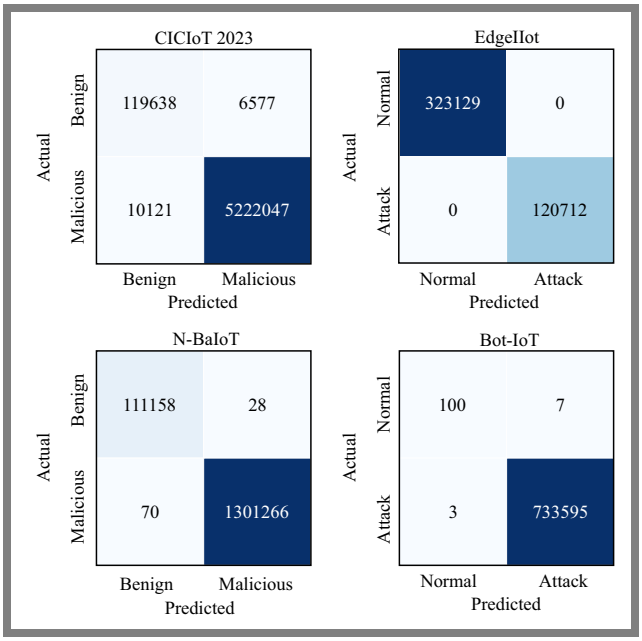

Fig. 3. Confusion matrices for binary anomaly detection across the four benchmark datasets. Each matrix shows the distribution of predicted versus actual classes (benign vs. malicious).

This reflects structural similarity in authentication-related behaviors, such as repeated log-in attempts over TCP, which the model conflates with other high-frequency traffic.

Bot-IoT and N-BaIoT maintain near-perfect scores across all categories, with dominant classes (DDoS, DoS, Mirai) classified with over 99% precision and recall. Confusion matrices highlight diagonal dominance for major categories, confirming the strength in distinguishing broad attack types through rate-driven features, e.g., packet-per-second rates, and flow duration.

5.3. Fine-grained Classification

Fine-grained classification (Tab. 5, Fig. 5) targets specific subtypes (DDoS UDP Flood, OS Fingerprinting). Although the weighted accuracy remains high (99.24% on CICIoT2023), the performance varies significantly for rare or overlapping

Tab. 3. Performance of the binary classification of the proposed model across four benchmark data sets.

Dataset	Class/metrics	Precision	Recall	F1 score	Support
CICIoT2023	Benign	92.00%	95.00%	93.00%	126 215
	Malicious	100.00%	100.00%	100.00%	5 232 168
	Accuracy			100.00%	
	Macro avg	96.00%	97.00%	97.00%	5 358 383
	Weighted avg	100.00%	100.00%	100.00%	5 358 383
N-BaIoT	Benign	100.00%	100.00%	100.00%	111 186
	Malicious	100.00%	100.00%	100.00%	1 301 336
	Accuracy			100.00%	
	Macro avg	100.00%	100.00%	100.00%	1 412 522
	Weighted avg	100.00%	100.00%	100.00%	1 412 522
Bot-IoT	Normal	97.1%	93.5%	95.2%	107
	Attack	100.00%	100.00%	100.00%	733 598
	Accuracy			100.00%	
	Macro avg	98.5%	96.7%	97.6%	733 705
	Weighted avg	99.3%	98.4%	98.8%	733 705
EdgeIIoT	Normal	100.00%	100.00%	100.00%	323 129
	Attack	100.00%	100.00%	100.00%	120 712
	Accuracy			100.00%	
	Macro avg	100.00%	100.00%	100.00%	443 841
	Weighted avg	100.00%	100.00%	100.00%	443 841

subcategories. Dominant subtypes like DDoS TCP Flood and Mirai achieve an F1 score, driven by distinctive volumetric patterns (sustained high packet rates, unique combinations of TCP flags).

However, minority classes such as XSS (12.18% recall) and Uploading attack (0% precision/recall) are systematically misclassified. In CICIoT2023, SqlInjection, CommandInjection, and BrowserHijacking are frequently conflated due to shared HTTP methods (POST) and common port usage, which the flow-level metadata cannot disentangle. The Bot-IoT OS fingerprint class is largely absorbed by the service scan and TCP categories, likely because the aggregate statistics do not capture TTL variations critical to fingerprinting. N-BaIoT TCP flag variants (ACK flood) exhibit 0% recall, indicating insufficient feature representation for flag-based distinctions, such as ACK-ratio thresholds.

5.4. Ablation Study

An ablation study in CICIoT2023 (Tab. 6) underscores the importance of architectural components in ensuring good performance. Removal of the shared encoder, a core element that enables MTL, degrades category classification by 10.41 percentage points and subcategory classification by 5.8 points. This highlights the necessity of shared representations to propagate discriminative features across hierarchical tasks.

The gating mechanism, which propagates features from binary tasks to category tasks, improves the category F1 score by approx. 4 points, while attention and batch normalization contribute approx. 3-point gains across multiclass tasks. These findings emphasize the interdependence of architectural elements in maintaining hierarchical consistency and mitigating the propagation of errors.

5.5. Baseline Comparison

Baseline comparisons (Tab. 7) further validate the model's superiority. Against classical methods like RF and Adaboost [10], multitask DNN achieves higher recall and F1 score, particularly in high-granularity settings. For example, in the 34-class classification on CICIoT2023, the model outperforms RF by 10 percentage points in the recall and 4 points in F1 score.

This gap widens with class imbalance and feature overlap, demonstrating the advantage in leveraging shared patterns across tasks to mitigate data scarcity in minority classes.

6. Discussion

The proposed lightweight multitask deep neural network (DNN) demonstrates robust performance across diverse IoT intrusion detection tasks, validating its suitability for real-time deployment in resource-constrained environments. Using shared representations across hierarchical classification levels, the model achieves high accuracy (up to 100% weighted F1 score) while maintaining computational efficiency. These results affirm the advantages of multitask learning in balancing generalization and specificity. However, limitations emerge in distinguishing rare or structurally similar subcategories, highlighting critical areas for improvement.

The model excels at identifying dominant attack patterns, particularly volumetric floods such as DDoS UDP_Flood, Mirai and protocol-driven anomalies, e.g., SYN floods. Across all datasets, these classes achieve near-perfect precision (over 99%) and recall ($> 99\%$), driven by rate-based features (flow duration, packet-per-second rates) and distinct TCP/UDP flag patterns. For example, N-BaIoT and Bot-IoT exhibit 100%

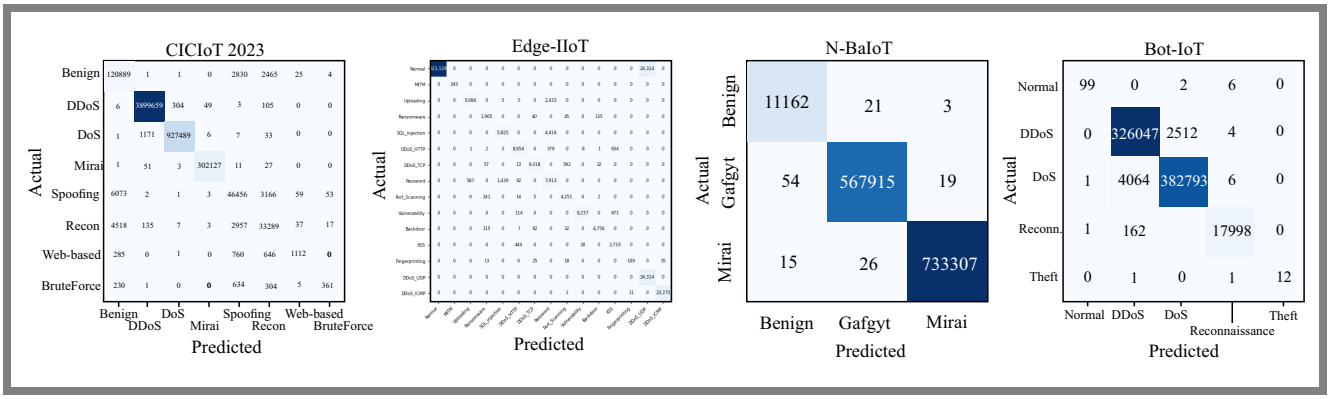

Fig. 4. Confusion matrices for coarse-grained attack classification across the four datasets: CICIoT2023, N-BaIoT, Bot-IoT, and Edge-IIoT.

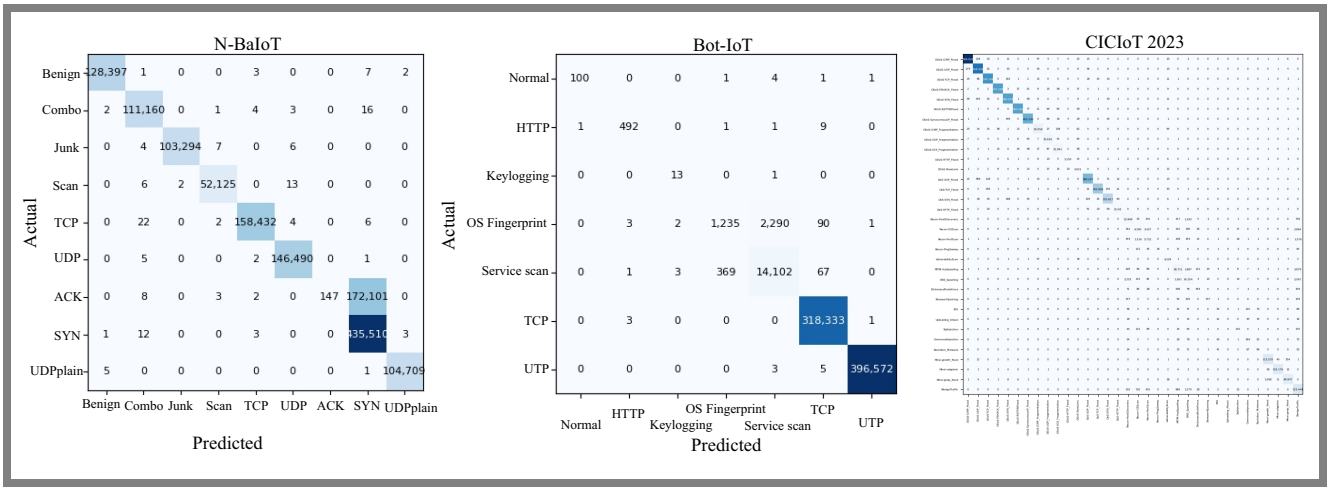

Fig. 5. Confusion matrices for fine-grained attack subtype classification across the CICIoT2023, N-BaIoT and Bot-IoT datasets.

binary classification accuracy, reflecting a flawless separation between benign traffic and large-scale attacks. Similarly, CICIoT2023 achieves 99.5% weighted accuracy in coarse-grained classification, with DDoS, DoS, and Mirai categories showing diagonal dominance in confusion matrices.

These successes stem from the model's ability to exploit temporal and volumetric signals, eg, burst traffic patterns, high packet rates, that distinguish dominant attacks from normal behavior. The shared encoder further enhances generalization by propagating discriminative features across tasks, as evidenced by the ablation study: removing the shared encoder degraded category classification by 10.41 percentage points.

Minority classes (XSS, BruteForce, SQL injection) suffer from poor recall ($< 40\%$ in CICIoT2023), exacerbated by two interrelated factors. First, severe data imbalance plagues these categories, with minority classes such as XSS (427 samples in CICIoT2023) outnumbered by dominant attacks by $1-3$ orders of magnitude. Second, feature ambiguity arises from shared protocol fields (e.g., HTTP POST methods, standard ports 80/443) and negligible inter-packet gaps, creating overlaps that obscure distinctions between classes.

For example, CICIoT2023's Web-based and BruteForce classes are frequently misclassified due to indistinguishable header-level statistics, even though precision remains above 90%.

In N-BaIoT, TCP flag variants (e.g. ACK floods) exhibit 0% recall, revealing a lack of explicit flag-based features, e.g. ACK-ratio thresholds. Similarly, Bot-IoT's OS Fingerprinting class (3 621 samples) is misclassified as Service Scan due to aggregate statistics failing to capture TTL and window-size variations. These errors underscore the model's inability to isolate subtle protocol behaviors when critical discriminative features are absent from the input representation.

Flow-level metadata lacks critical granular cues (e.g., payload entropy, token sequences) for low-volume attacks. In CICIoT2023, SQL Injection, Command Injection, and Browser Hijacking are conflated due to shared HTTP methods and port usage, achieving only $19-43\%$ recall. This limitation highlights the inherent constraints of header-only analysis in distinguishing attacks that rely on nuanced payload content or application-layer logic.

The ablation study underscores the importance of key architectural components. The gating mechanism, which propagates binary-to-category features, improves F1 scores by 4 points, while attention contribute 3-point gains in multiclass tasks by enhancing feature adaptability.

In the 34-class CICIoT2023 classification, DNN outperforms RF by 10 percentage points in recall and 4 points in F1 score, demonstrating the advantage in mitigating data scarcity through shared representations.

Tab. 4. Coarse-grained classification performance on datasets.

Class/metrics	Precision	Recall	F1 score	Support
CICIoT2023				
Benign	92.00%	96.00%	94.00%	126 215
DDoS	100.00%	100.00%	100.00%	3 900 126
DoS	100.00%	100.00%	100.00%	928 707
Mirai	100.00%	100.00%	100.00%	302 220
Spoofing	87.00%	83.00%	85.00%	55 813
Recon	83.00%	81.00%	82.00%	40 963
Web-based	90.00%	40.00%	55.00%	2804
BruteForce	83.00%	24.00%	37.00%	1535
Accuracy		99.50%		
Macro avg	92.00%	78.00%	82.00%	5 358 383
Weighted avg	99.00%	99.00%	99.00%	5 358 383
EdgeIIoT				
Normal	100%	100%	100%	323 129
MITM	100%	100%	100%	243
Uploading	90%	68%	77%	7527
Ransomware	82%	90%	86%	2185
SQL_injection	80%	57%	67%	10 241
DDoS_HTTP	93%	90%	91%	9982
DDoS_TCP	99%	93%	96%	10 012
Password	52%	79%	63%	10 031
Port_Scanning	86%	94%	90%	4513
Vulnerability_scanner	100%	92%	96%	10 022
Backdoor	97%	96%	96%	4972
XSS	68%	85%	75%	3183
Fingerprinting	91%	55%	68%	200
DDoS_UDP	100%	100%	100%	24 314
DDoS_ICMP	100%	100%	100%	23 287
Accuracy		97.13%		
Macro avg	89%	87%	87%	443 841
Weighted avg	98%	97%	97%	443 841
Bot-IoT				
Normal	98.0%	92.5%	95.2%	107
DDoS	98.9%	99.3%	99.1%	385 309
DoS	99.2%	98.8%	99.0%	330 112
Reconnaissance	99.9%	99.1%	99.5%	18 163
Theft	100.0%	85.7%	92.3%	14
Macro avg	99.2%	95.1%	97.0%	733 705
Weighted avg	99.2%	97.1%	98.1%	1 467 410

7. Conclusions

Comprehensive experiments on four benchmark data sets demonstrate the robustness of the proposed model across multiple classification levels. Despite its strengths, HC-MTDNN encounters challenges with fine-grained detection of structurally similar or low-prevalence attacks, such as XSS and SQL injection. This low effectiveness is not merely a limitation in feature discriminability or class imbalance, but a direct consequence of the model's reliance on flow-level metadata, excluding payload analysis. This is a conscious design trade-off to maintain the lightweight nature, enabling deployment in resource-constrained IoT settings where full packet inspection may be infeasible due to encryption, privacy concerns, or computational overhead.

Future work will be focused on augmenting the feature space with lightweight payload-derived statistics (e.g., entropy metrics, token frequencies), temporal behavior modeling, and

Tab. 5. Coarse-grained classification performance.

Class/metrics	Precision	Recall	F1 score	Support
CICIoT2023				
Benign	92.00%	96.00%	94.00%	126 215
DDoS-ICMP_Flood	99.96%	99.96%	99.96%	826914
DDoS-UDP_Flood	99.85%	99.94%	99.90%	618833
DDoS-TCP_Flood	99.89%	99.92%	99.91%	516498
DDoS-PSHACK_Flood	99.98%	99.96%	99.97%	471782
DDoS-SYN_Flood	99.83%	99.90%	99.87%	466143
DDoS-RSTFINFlood	99.98%	99.93%	99.96%	463864
DDoS-SynonymIP_Flood	99.92%	99.87%	99.89%	412675
DDoS-ICMP_Fragmentation	99.53%	99.26%	99.40%	52443
DDoS-UDP_Fragmentation	99.01%	99.38%	99.20%	32820
DDoS-ACK_Fragmentation	98.97%	99.16%	99.07%	32211
DDoS-HTTP_Flood	98.16%	97.95%	98.05%	3216
DDoS-SlowLoris	87.02%	96.11%	91.34%	2727
DoS-UDP_Flood	99.90%	99.83%	99.87%	380875
DoS-TCP_Flood	99.97%	99.82%	99.90%	306346
DoS-SYN_Flood	99.84%	99.76%	99.80%	233180
DoS-HTTP_Flood	98.69%	98.03%	98.36%	8306
Recon-HostDiscovery	76.94%	83.96%	80.30%	15479
Recon-OSScan	62.74%	38.84%	47.98%	11304
Recon-PortScan	56.25%	59.77%	57.96%	9590
Recon-PingSweep	73.08%	7.09%	12.93%	268
VulnerabilityScan	94.47%	97.27%	95.85%	4322
MITM-ArpSpoofing	88.89%	81.47%	85.02%	35240
DNS_Spoofing	72.19%	69.28%	70.71%	20573
DictionaryBruteForce	68.48%	29.58%	41.31%	1535
BrowserHijacking	81.07%	20.33%	32.50%	674
XSS	29.71%	12.18%	17.28%	427
Uploading_Attack	0.00%	0.00%	0.00%	137
SqlInjection	55.56%	19.13%	28.46%	575
CommandInjection	53.55%	43.21%	47.83%	611
Backdoor_Malware	39.02%	25.26%	30.67%	380
Mirai-greeth_flood	99.00%	99.63%	99.31%	113958
Mirai-udpplain	99.94%	99.94%	99.94%	102242
Mirai-greip_flood	99.52%	98.64%	99.08%	86020
BenignTraffic	90.74%	97.01%	93.77%	126215
Accuracy	99.24%	99.24%	99.24%	
Macro avg	82.99%	75.63%	77.21%	5358383
Weighted avg	99.21%	99.24%	99.21%	5358383
Bot-IoT				
normal	99.0%	93.5%	96.2%	107
HTTP	98.6%	97.6%	98.1%	504
Keylogging	72.2%	92.9%	81.3%	14
OS_Fingerprint	76.9%	34.1%	47.3%	3621
Service_Scan	86.0%	97.0%	91.1%	14542
TCP	99.9%	100.0%	100.0%	318337
UDP	100.0%	100.0%	100.0%	396580
Macro avg	90.4%	87.9%	87.7%	733705
Weighted avg	95.0%	93.7%	93.6%	1467410
N-BaIoT				
Benign	100.00%	100.00%	100.00%	128410
Combo	100.00%	100.00%	100.00%	111186
Junk	100.00%	100.00%	100.00%	103311
Scan	100.00%	100.00%	100.00%	52146
TCP	100.00%	100.00%	100.00%	158466
UDP	100.00%	100.00%	100.00%	146498
ACK	100.00%	0.00%	0.00%	172261
SYN	72.00%	100.00%	83.00%	435529
udpplain	100.00%	100.00%	100.00%	104715
Accuracy		87.80%		
Macro avg	97.00%	89.00%	87.00%	1412522
Weighted avg	91.00%	88.00%	83.00%	1412522

Tab. 6. Ablation study results in the CICIoT2023 dataset, evaluating the contribution of key architectural components to the proposed model.

Variant	Binary		Category		Subcategory	
	F_1 [%]	AUC [%]	F_1 [%]	Δ [pts]	F_1 [%]	Δ [pts]
HC-MTDNN	99.84	99.95	81.53	—	77.21	—
No attention	99.77	99.92	78.48	–3.05	74.06	–3.15
No batch normalization	99.78	99.92	77.99	–3.54	74.00	–3.21
No gating	99.76	99.92	77.43	–4.10	73.72	–3.49
No gating, no attention	99.80	99.93	78.93	–2.60	73.86	–3.35
No shared encoder	99.72	99.86	71.12	–10.41	71.41	–5.80

sequence-aware components. These additions could improve classification fidelity without sacrificing real-time capability. Addressing encrypted traffic detection through enhanced metadata analysis and equipping the model with mechanisms for continuous learning and uncertainty estimation would further expand its applicability.

References

[1] R. Chataut, A. Phoummalayvane, and R. Akl, "Unleashing the Power of IoT: A Comprehensive Review of IoT Applications and Future Prospects in Healthcare, Agriculture, Smart Homes, Smart Cities, and Industry 4.0", *Sensors*, vol. 23, art. no. 7194, 2023 (https://doi.org/10.3390/s23167194).

[2] F. Nie, W. Liu, G. Liu, and B. Gao, "M2VT-IDS: A Multi-task Multi-view Learning Architecture for Designing IoT Intrusion Detection System", *Internet of Things*, vol. 25, art. no. 101102, 2024 (https://doi.org/10.1016/j.iot.2024.101102).

[3] P.S. Bangare and K.P. Patil, "Security Issues and Challenges in Internet of Things (IoT) System", *2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, Greater Noida, India, 2022 (https://doi.org/10.1109/ICACITE53722.2022.9823709).

[4] A.A. Rokhade *et al.*, "Anomaly Detection for IoT Security: Comprehensive Survey", *2023 International Conference on Advances in Electronics, Communication, Computing and Intelligent Information Systems (ICAECIS)*, Bangalore, India, 2023 (https://doi.org/10.1109/ICAECIS58353.2023.10170192).

[5] M. Antonakakis *et al.*, "Understanding the Mirai Botnet", *26th USENIX Conference on Security Symposium (SEC'17)*, Vancouver, Canada, 2017 (https://doi.org/10.13140/RG.2.2.24145.54885).

[6] H. Rhachi, Y. Balboul, and A. Bouayad, "Enhanced Anomaly Detection in IoT Networks Using Deep Autoencoders with Feature Selection Techniques", *Sensors*, vol. 25, art. no. 3150, 2025 (https://doi.org/10.3390/s25103150).

[7] A.G. Kumar, A. Rastogi, and V. Ranga, "Evaluation of Different Machine Learning Classifiers on New IoT Dataset CICIoT2023", *2024 International Conference on Intelligent Systems for Cybersecurity (ISCS)*, Gurugram, India, 2024 (https://doi.org/10.1109/ISCS61804.2024.10581375).

[8] K.R. Narayan *et al.*, "IIDS: Design of Intelligent Intrusion Detection System for Internet-of-Things Applications", *2023 IEEE 7th Conference on Information and Communication Technology (CICT)*, Jabalpur, India, 2023 (https://doi.org/10.1109/CICT59886.2023.10455720).

[9] R. Caruana, "Multitask Learning", *Machine Learning*, vol. 28, pp. 41–75, 1997 (https://doi.org/10.1023/A:1007379606734).

[10] E.C.P. Neto *et al.*, "CICIOT2023: A Real-time Dataset and Benchmark for Large-scale Attacks in IoT Environment", *Sensors*, vol. 23, art. no. 5941, 2023 (https://doi.org/10.3390/s23135941).

[11] B. Meidan *et al.*, "N-BaIoT Dataset to Detect IoT Botnet Attacks", IEEE Pervasive Computing, vol. 17, pp. 12–22, 2018 (https://doi.org/10.1109/MPRV.2018.03367731).

[12] N. Koroniotis *et al.*, "Towards the Development of Realistic Botnet Dataset in the Internet of Things for Network Forensic Analytics: Bot-IoT Dataset", *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2018 (https://doi.org/10.1016/j.future.2019.05.041).

[13] M.A. Ferrag *et al.*, "Edge-IIoTSET: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning", *IEEE Access*, vol. 10, pp. 40281–40306, 2022 (https://doi.org/10.1109/ACCESS.2022.3165809).

[14] Imran *et al.*, "Realtime Feature Engineering for Anomaly Detection in IoT Based MQTT Networks", *IEEE Access*, vol. 12, pp. 25700–25718, 2024 (https://doi.org/10.1109/ACCESS.2024.3363889).

[15] Z. Hafezian, M. Naderan, and M. Jaderyan, "A Machine Learning-based Approach for Multi-class Intrusion Detection and Classification in IoT Using CICIoT2023 Dataset", *2024 11th International Symposium on Telecommunications (IST)*, Tehran, Iran, 2024 (https://doi.org/10.1109/IST64061.2024.10843502).

[16] A. Hajjouz and E. Avksentieva, "Optimizing Intrusion Detection for DoS, DDoS, and Mirai Attacks Subtypes Using Hierarchical Feature Selection and CatBoost on the CICIoT2023 Dataset", *Data and Metadata*, vol. 3, art. no. 577, 2024 (https://doi.org/10.56294/dm2024577).

[17] N. Thereza and K. Ramli, "Development of Intrusion Detection Models for IoT Networks Utilizing CICIoT2023 Dataset", *2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)*, Bali, Indonesia, 2023 (https://doi.org/10.1109/ICON-SONICS59898.2023.10435006).

[18] C. Eunaicy, c. Jayapratha, and H.S. Hemachitra, "IoT Guardian: A Novel Feature Discovery and Cooperative Game Theory Empowered Feature Selection with ML Model for IoT Threats and Attack Detection", *International Journal of Computer Networks and Communications*, vol. 16, pp. 25–42, 2024 (https://doi.org/10.5121/ijcnc.2024.16202).

[19] J.R.K. Rajasekaran, B. Natarajan, and A. Pahwa, "Modified Matrix Completion-based Detection of Stealthy Data Manipulation Attacks in Low Observable Distribution Systems", *IEEE Transactions on Smart Grid*, vol. 14, pp. 4851–4862, 2023 (https://doi.org/10.1109/TSG.2023.3266834).

[20] H. Dong and I. Kotenko, "An Autoencoder-based Multi-task Learning for Intrusion Detection in IoT Networks", *2023 IEEE Ural-Siberian Conference on Biomedical Engineering, Radioelectronics and Information Technology (USBEREIT)*, Yekaterinburg, Russia, 2023 (https://doi.org/10.1109/USBEREIT58508.2023.10158807).

[21] Z. Wang *et al.*, "A Lightweight Intrusion Detection Method for IoT Based on Deep Learning and Dynamic Quantization", *PeerJ Computer Science*, vol. 9, art. no. 1569, 2023 (https://doi.org/10.7717/peerj-cs.1569).

[22] J. Chen, J. Xiao, and J. Xu, "VGGIncepNet: Enhancing Network Intrusion Detection and Network Security Through Non-image-to-image Conversion and Deep Learning", *Electronics*, vol. 13, art. no. 3639, 2024 (https://doi.org/10.3390/electronics13183639).

[23] A. Hinojosa and N.E. Majd, "Edge Computing Network Intrusion Detection System in IoT Using Deep Learning", *2024 33rd International Conference on Computer Communications and Networks (ICCCN)*, Kailua-Kona, USA, 2024 (https://doi.org/10.1109/ICCCN61486.2024.10637611).

[24] T. Hasan and S. Tasnim, "Multidimensional Feature Learning Enhancement in IoT Intrusion Detection: An Adaptive Cost-sensitive Autoencoder and Weighted Ensemble Approach", *2024 IEEE 10th World Forum on Internet of Things (WF-IoT)*, Ottawa, Canada, 2024 (https://doi.org/10.1109/WF-IoT62078.2024.10811174).

Tab. 7. Baseline results for binary, 8-class, and 34-class classification.

Metric	Logistic regression	Perceptron	Adaboost	RF	DNN
	2-class (binary)				
Accuracy	0.9890	0.9818	0.9959	0.9968	0.9944
Recall	0.8904	0.7970	0.9473	0.9652	0.9333
Precision	0.8632	0.8254	0.9656	0.9654	0.9476
F1 score	0.8763	0.8105	0.9563	0.9653	0.9403
	8-class				
Accuracy	0.8317	0.8663	0.3514	0.9944	0.9911
Recall	0.6961	0.6591	0.4878	0.9100	0.9066
Precision	0.5124	0.5239	0.4649	0.7054	0.6794
F1 score	0.5394	0.5551	0.3687	0.7193	0.6973
	34-class				
Accuracy	0.8023	0.8196	0.6079	0.9916	0.9861
Recall	0.5952	0.5075	0.6077	0.8316	0.7319
Precision	0.4868	0.4546	0.4796	0.7045	0.6653

[25] S. Yan, H. Han, X. Dong, and Z. Xu, "Lightweight Deep Learning Method Based on Group Convolution: Detecting DDoS Attacks in IoT Environments", *2024 10th International Symposium on System Security, Safety, and Reliability (ISSSR)*, Xiamen, China, 2024 (https://doi.org/10.1109/ISSSR61934.2024.00027).

[26] M.A. Elaziz, I.A. Fares, and A.O. Aseeri, "CKAN: Convolutional Kolmogorov-Arnold Networks Model for Intrusion Detection in IoT Environment", *IEEE Access*, vol. 12, pp. 134837–134851, 2024 (https://doi.org/10.1109/ACCESS.2024.3462297).

[27] H. Kamal and M. Mashaly, "Enhanced Hybrid Deep Learning Models-based Anomaly Detection Method for Two-stage Binary and Multi-class Classification of Attacks in Intrusion Detection Systems", *Algorithms*, vol. 18, 2025 (https://doi.org/10.3390/a18020069).

[28] Z. Ahmad *et al.*, "Anomaly Detection Using Deep Neural Network for IoT Architecture", *Applied Sciences*, vol. 11, art. no. 7050, 2021 (https://doi.org/10.3390/app11157050).

[29] A.N. Gummadi, J.C. Napier, and M. Abdallah, "XAI-IoT: An Explainable AI Framework for Enhancing Anomaly Detection in IoT Systems", *IEEE Access*, vol. 12, pp. 71024–71054, 2024 (https://doi.org/10.1109/ACCESS.2024.3402446).

[30] E. Altulaihan, M.A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms", *Sensors*, vol. 24, art. no. 713, 2024 (https://doi.org/10.3390/s24020713).

[31] Z. Zulfiqar *et al.*, "DeepDetect: An Innovative Hybrid Deep Learning Framework for Anomaly Detection in IoT Networks", *Journal of Computational Science*, vol. 83, art. no. 102426, 2024 (https://doi.org/10.1016/j.jocs.2024.102426).

[32] A. Alrefaei and M. Ilyas, "Using Machine Learning Multiclass Classification Technique to Detect IoT Attacks in Real Time", *Sensors*, vol. 24, art. no. 4516, 2024 (https://doi.org/10.3390/s24144516).

[33] Z. Alwaisi, T. Kumar, E. Harjula, and S. Soderi, "Securing Constrained IoT Systems: A Lightweight Machine Learning Approach for Anomaly Detection and Prevention", *Internet of Things*, vol. 28, art. no. 101398, 2024 (https://doi.org/10.1016/j.iot.2024.101398).

[34] I. Ullah and Q.H. Mahmoud, "Design and Development of RNN Anomaly Detection Model for IoT Networks", *IEEE Access*, vol. 10, pp. 62722–62750, 2022 (https://doi.org/10.1109/ACCESS.2022.3176317).

[35] M.M. Khan and M. Alkhathami, "Anomaly Detection in IoT-based Healthcare: Machine Learning for Enhanced Security", *Scientific Reports*, vol. 14, art. no. 5872, 2024 (https://doi.org/10.1038/s41598-024-56126-x).

[36] S.-M. Tseng, Y.-Q. Wang, and Y.-C. Wang, "Multi-class Intrusion Detection Based on Transformer for IoT Networks Using CIC-IoT-2023 Dataset", *Future Internet*, vol. 16, art. no. 284, 2024 (https://doi.org/10.3390/fi16080284).

[37] R. Zhao *et al.*, "A Novel Intrusion Detection Method Based on Lightweight Neural Network for Internet of Things", *IEEE Internet of Things Journal*, vol. 9, pp. 9960–9972, 2022 (https://doi.org/10.1109/JIOT.2021.3119055).

[38] M. Verkerken *et al.*, "A Novel Multi-stage Approach for Hierarchical Intrusion Detection", *IEEE Transactions on Network and Service Management*, vol. 20, pp. 3915–3929, 2023 (https://doi.org/10.1109/TNSM.2023.3259474).

[39] A. Momand, S.U. Jan, and N. Ramzan, "ABCNN-IDS: Attention-based Convolutional Neural Network for Intrusion Detection in IoT Networks", *Wireless Personal Communications*, vol. 136, pp. 1981–2003, 2024 (https://doi.org/10.1007/s11277-024-11260-7).

[40] S. Ali *et al.*, "Effective Multitask Deep Learning for IoT Malware Detection and Identification Using Behavioral Traffic Analysis", *IEEE Transactions on Network and Service Management*, pp. 1199–1209, 2022 (https://doi.org/10.1109/TNSM.2022.3200741).

Mohamed Amine Beghoura, Ph.D.
Department of Computer Science
https://orcid.org/0000-0002-8355-8071
E-mail: mohamedamine.beghoura@univ-bba.dz
University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Younes Belouche, Ph.D. student
Department of Computer Science
https://orcid.org/0009-0001-7809-2561
E-mail: younes.belouche@univ-bba.dz
University of Mohamed El Bachir El Ibrahimi, Bordj Bou Arreridj, Algeria
https://www.univ-bba.dz

Deep Learning-based Compensation for Doppler Shifts in Hybrid Beamforming for mmWave Communication

Kartik Ramesh Patel and Sanjay Dasrao Deshmukh

MCT's Rajiv Gandhi Institute of Technology, Mumbai, India

https://doi.org/10.26636/jtit.2025.4.2349

Abstract — Millimeter-wave (mmWave) communication is a key enabler of 5G and future wireless systems, providing vast bandwidth for high-speed data transfers. However, high user mobility leads to significant Doppler shifts, which can severely degrade the performance of beamforming – an essential technology for mmWave systems. The traditional hybrid beamforming (HBF) technique faces challenges in adapting to rapid channel variations caused by Doppler effects. Therefore, this paper introduces a deep learning framework to mitigate Doppler-induced channel distortions in hybrid beamforming. We propose a long-short-term memory (LSTM)-based neural network that predicts Doppler shifts and dynamically adjusts the hybrid beamforming vectors to compensate for these variations. This approach proactively addresses channel distortion, enhancing both spectral and energy efficiency. The simulation results and the performance comparison of proposed model against conventional beamforming and state-of-the-art techniques demonstrate the superiority of deep learning-based solution in maintaining robust communication links under high-mobility conditions, showcasing its potential to improve performance in next-generation wireless networks.

Keywords — Doppler shift, hybrid beamforming LSTM, mmWave, spectral efficiency

1. Introduction

The huge growth in mobile data traffic, driven by applications such as ultra-high definition video, autonomous systems, and the Internet of Things (IoT) [1] has resulted in demand for unprecedented data rates and ultra-low communication latency. This demand has propelled wireless communication into the millimeter-wave (mmWave) (30 – 300 GHz) spectrum, [2] a frontier defined by its vast contiguous bandwidth.

However, the mmWave spectrum presents physical challenges, such as severe loss of path loss and susceptibility to blockage [3]. To overcome these limitations, large-scale antenna arrays are employed to achieve high-gain beamforming, focusing signal energy into narrow, directional beams to establish and maintain a robust communication link. To manage costs, hardware complexity and power consumption with fully digital beamforming, hybrid beamforming (HBF) has emerged as a consensus energy-efficient architecture [4]. By partitioning the beamforming task between a high-dimensional analog domain (using phase shifters) and a low-dimensional digital domain (at baseband), HBF provides a balance of array gain and system cost.

Despite its architectural merits, the efficacy of HBF depends on accurate and real-time channel state information (CSI). In high-mobility scenarios, such as vehicle-to-everything (V2X) communication, high-speed rail, and drone networks, this contingency becomes a bottleneck [5]. The movement between the transmitter and the receiver causes significant Doppler shifts manifesting as rapid phase variations in the channel [6]. This phenomenon causes the channel to decompose over time, invalidating the "quasi-static" assumption upon which conventional, reactive beam-tracking algorithms and codebook-based HBF solutions are built. It leads to severe performance degradation, inter-channel interference, and even link failure.

Coherence time shrinks at highway speeds, making reactive beam updates insufficient [2], while classical Kalman filter (KF) and extended Kalman filter (EKF) [7] trackers update angles and recent deep learning works often predict beam indices, they do not incorporate the hybrid analog-digital pair at each slot from a predicted complex channel, thus leaving a gap in Doppler-aware HBF design.

Researchers identified the time-series nature of the problem, applying recurrent neural networks (RNN) [8] and their variants, such as LSTM and gated recurrent units (GRU), to predict future channel states [9].

Despite that, the research gap persists. Most current works focus on the prediction of the full-dimensional, unconstrained CSI matrix. There is still a significant disconnect between this high-dimensional prediction and its practical, real-time application within the constrained HBF architecture. Recently, research has moved to exploring alternatives, such as deep reinforcement learning (DRL) for policy-based beam control [10] and transformer-based models for long-range temporal dependency. Although promising, DRL models can suffer from training instability, and transformers carry a significant computational overhead.

Conventional hybrid beamforming techniques, which rely on the assumption of a quasi-static channel, fail in high-mobility mmWave environments due to rapid channel decorrelation [11]. Although recent deep learning models demonstrate the ability to predict channel variations, there is a re-

search gap in developing a framework that efficiently and proactively integrates the predictive information into the physically constrained and latency-constrained hybrid beamforming architecture in order to maintain robust, high-throughput communication links [12].

In such a context, we propose a pipeline type, where an LSTM predicts $\widehat{\mathbf{H}}(t+1)$ from the past K channels, and next that prediction drives analog and digital stages under constant module constraints.

Recent works on mobility-robust mmWave links mainly tracks beams with state-space filters, i.e. KF/EKF or learns beam indices directly with neural networks, often without redesigning the full hybrid analog-digital chain under motion.

The contribution of this paper can be summarized as follows:

- The impact of high Doppler shifts on the time-varying mmWave channel is modeled and effect quantified on a conventional hybrid beamforming system.
- The predict-then-design deep learning framework is used, centered on an LSTM-based predictive engine, which proactively forecasts the evolution of the mmWave channel and directly computes the required compensatory HBF vectors.
- A dynamic HBF compensation algorithm is implemented that translates the LSTM predictive output into real-time, constrained analog and digital beamforming commands.
- A comprehensive simulation-based performance evaluation is presented, benchmarking the proposed framework against conventional (extended Kalman filter-based) and orthogonal matching pursuit (OMP) methods in terms of spectral efficiency, energy efficiency, and link robustness under various high-mobility scenarios.

The paper is organized as follows. Section 2 reviews related work. Section 3 presents the method concerned, while Section 4 discusses the simulation results and evaluates the performance. Section 5 concludes the paper.

2. Related Work

The authors of [7] provide a insight into extended Kalman filters, which are solution for non-linear tracking problems. In this context, EKF works in a two-step predict-update loop. It uses the system's mobility model to predict the next channel state and the actual received pilot signal to correct the prediction. Particle filters are a more robust, non-parametric alternative to EKF. They represent the channel state not as a single estimate with covariance (like EKF), but as a cloud of thousands of weighted particles [7].

In both approaches, the computational complexity is extremely high. EKFs involve large matrix inversions at every time step, and particle filters are more brittle, relying on an accurate pre-defined mathematical model of the user's movement, which is not suitable in highly dynamic environments.

A hierarchical beam sweeping method is proposed in [12], where instead of estimating the full channel matrix, the system simply tries to find the best pre-set beam. A hierarchical search avoids testing all beams. First, it uses wide beams to find the general direction, and then zooms in with progressively narrow beams. Unfortunately, such methods are difficult to adapt. A high-speed user can travel out the optimal beam in the time it takes the algorithm to complete its sweep-and-search process. Hence, this method is fundamentally reactive. It can only fix a beam misalignment after it has already occurred and caused performance degradation.

The importance of using the HBF architecture is explained in [4]. The analog beamformer has a large matrix of simple, low-cost phase shifters but does the coarse beam steering, while the digital beamforming has an expensive single antenna element, dedicated, and power-hungry radio frequency chain. Therefore, the hybrid beamforming architecture is considered a compromise between analog and digital architecture with few RF chains.

The HBF architecture using codebook-based precoding is described in [13]. The codebook is a predefined set of high-performance analog and digital beamforming vectors. The system tracking job is reduced by selecting the best index of this codebook. The problem with this method lies in the codebooks designed for static and slowly varying channels. The entire codebook is generated offline, assuming the channel is stable. The Doppler effect breaks this assumption. The true optimal beam for high-speed user will likely lie in between any two precalculated beams in the codebook, leading to a constant, suboptimal quantization error.

The authors in [10] mentioned that modern wireless channels are so complex (with blocks, reflections, and mobility) that model-based approaches from [7], [12], are no longer feasible. They conclude that machine learning and deep learning act as universal function approximates to learn the complex mapping from received pilots to optimal beam directly from data, without needing a perfect mathematical model.

Paper [6] is the first attempt to apply deep learning to the problem. A convolutional neural network (CNN) is designed to treat the channel matrix $\mathbf{H}(t)$ as an image. CNNs are good at finding spatial features with the ability to detect, in the channel matrix, patterns such as dominant paths and their angles to determine the best beam at that instant. The downside of this method is that CNNs are not inherently designed to capture temporal dependencies. With no memory of the past, a CNN cannot see the user movement trend and, therefore, cannot predict the future.

The method of time series forecasting based on the requirement of the dynamic wireless channel is presented in [8] and [9], which is exactly why RNNs like LSTM and GRU are built. Unlike CNN, the LSTM has a memory that allows it to process a sequence of past channels to learn the underlying dynamics of the user's motion. The goal of these methods is purely prediction-oriented and they prove to be highly accurate [9].

The critical gap in the methods described in [7] and [8] is that they do not specifically address the integration of these predictions into the hybrid beamforming framework for Doppler mitigation. Both end at the prediction of $\widehat{\mathbf{H}}(t+1)$, but they do not provide an answer as to what is going to be

Tab. 1. Related literature vs. proposed method.

Approach	Predicts/tracks	Uses prediction to design both analog and digital HBF each slot	Mobility focus/Doppler	Remarks
EKF beam tracking [7]	Angles/kinematics	No	Vehicular EKF with Jacobians	Model-driven; low complexity; sensitive to mismatch
DL beam tracking [6]	Beam index	No	Mobility under sounding	High accuracy; not coupled to the hybrid precoder/combiner
Sub-6 mmWave beam/blockage [8]	Beam/blockage	No	Mobility/robustness	Multiband features; no HBF co-design
DL-HBF surveys/reviews [11]	–	Varies	Discuss challenges	Surveys DL for HBF; limited Doppler-aware co-design exemplars
This work (LSTM-HBF)	Channel (complex)	Yes	Explicit Doppler via sequence prediction	Adds the SE-loss bound, complexity, and full reproducibility

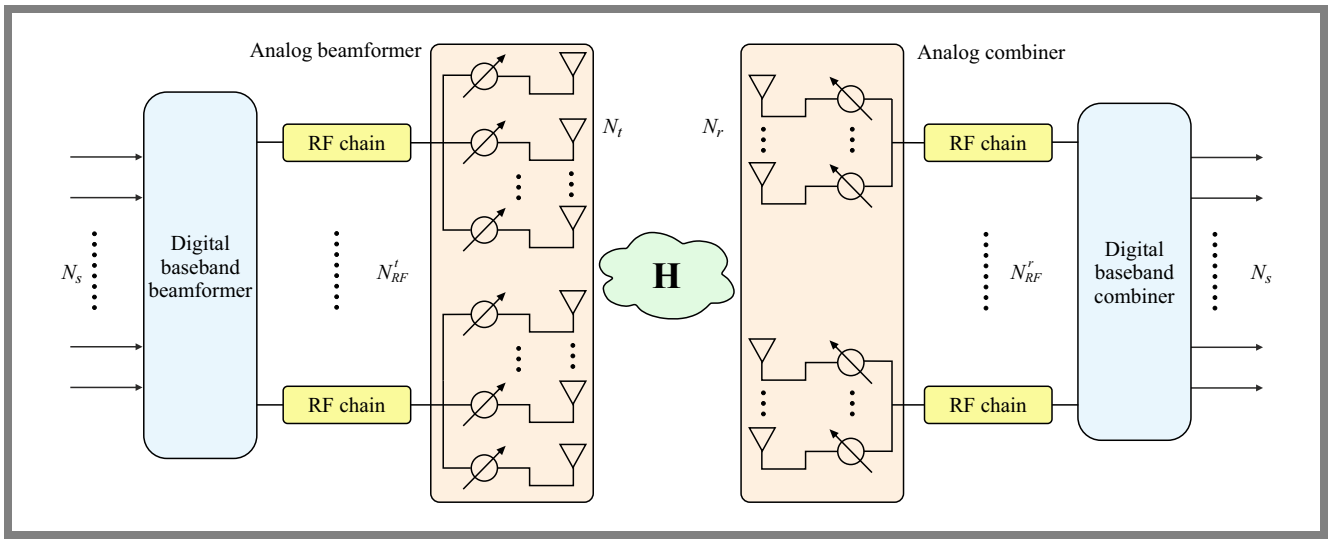

Fig. 1. Hybrid analog-digital architecture with N_t antennas and N_{RF}^t RF chains at BS and N_r, N_{RF}^r at UE side.

done next. There is a gap in how to use this predicted full-digital matrix $\widehat{\mathbf{H}}(t+1)$ to calculate constrained analog $\mathbf{F}_{RF}$ and digital $\mathbf{F}_{BB}$ matrices. These calculations have to be done quickly enough to proactively mitigate the Doppler effect.

The proposed method is based on these foundations, but it is distinct in its focus on proactively mitigating the Doppler effect in hybrid beamforming using a predictive LSTM model. It aims to bridge the gap between deep learning-based channel prediction and practical hybrid beamforming design for high-mobility mmWave systems. Table 1 shows the comparison of the proposed method with other articles.

3. Research Methodology

The proposed system consists of a massive mmWave MIMO block with a hybrid beamforming architecture, as shown in Fig. 1. The time-varying channel $\mathbf{H}(t)$ is modeled as a sum of the multiple paths (multipath fading). Each path is characterized by a Doppler shift $f_{D,l}$ which accounts for the relative motion of the UE and the BS. The Doppler shift $f_{D,l}$ for the l-th path is modeled using relative velocity v and angle of arrival θ_i. This is important for high-mobility scenarios, where the Doppler shift significantly impacts the channel. Matrices $\mathbf{F}_{RF}$ (analog precoder) and $\mathbf{W}_{RF}$ (analog combiner) are based on the SVD of the time-varying channel matrix to align the strongest eigen modes.

3.1. System Model

We consider a single-cell downlink scenario with a base station (BS) equipped with a uniform linear array (ULA) N_{BS} or N_t antenna. The mobile user (UE) has a ULA of N_{UE} or N_r antennas. BS uses a hybrid beamforming structure with N_{RF} RF chains, where $N_{RF} \ll N_{BS}$. Let N_t and N_r denote the numbers of transmit and receive antennas, N_{RF}^t and N_{RF}^r the number of RF chains. The N_s number of data streams, where $N_s \leqslant \min N_{RF}^t$.

The analog precoder/combiner is defined as $\mathbf{F}_{RF} \in \mathbb{C}^{N_t \times N_{RF}^t}$, $\mathbf{W}_{RF} \in \mathbb{C}^{N_r \times N_{RF}^r}$ with constant-modulus en-

tries:

$$\left|\left[F_{\mathrm{RF}}\right]_{ij}\right| = \frac{1}{\sqrt{N_t}}, \quad \left|\left[W_{\mathrm{RF}}\right]_{ij}\right| = \frac{1}{\sqrt{N_r}}.$$

The digital baseband precoder/combiner is $\mathbf{F}_{BB} \in \mathbb{C}^{N_t \times N_{RF}^t}$, $\mathbf{W}_{BB} \in \mathbb{C}^{N_{RF}^r \times N_s}$ and the transmit symbol vector $\boldsymbol{s} \in \mathbb{C}^{N_s}, \mathbb{E}\left[ss^H\right] = \frac{P}{N_s} I$ with the noise $n \sim \mathcal{CN}\left(0, \sigma_n^2 I\right)$. The array response vector for a ULA can be defined as:

$$\mathbf{a}_N(\theta) = \frac{1}{\sqrt{N}} \left[1, e^{j\pi d \sin\theta}, e^{j2\pi d \sin\theta}, \ldots, e^{j\pi(N-1)\sin\theta}\right]^T, \tag{1}$$

where N is the number of antennas and θ is the physical angle of departure or arrival.

Equation (1) defines the response vector for a ULA with N elements, where each element of the array introduces a phase change relative to others. The phase change is proportional to the distance between the antenna's elements, the wavelength of the signal, and the angle of arrival or departure.

The antennas are assumed to be in a uniform linear configuration, which is typical for MIMO systems, and the phase shifts depend on the angle of arrival (or departure) relative to the array axis.

The signal received at the UE can be modeled as:

$$y(t) = \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \mathbf{H}(t) \mathbf{F}_{RF} \mathbf{F}_{BB} \boldsymbol{s}(t) + \mathbf{W}_{BB}^H \mathbf{W}_{RF}^H \boldsymbol{n}(t), \tag{2}$$

where:

- $\mathbf{H}(t) \in \mathbb{C}^{N_r \times N_t}$ is the time-varying mmWave channel matrix,
- $\mathbf{F}_{RF} \in \mathbb{C}^{N_{BS} \times N_{RF}}$ and $\mathbf{F}_{BB} \in \mathbb{C}^{N_{RF} \times N_s}$ are the analog and digital precoders at the BS, respectively, where N_s is the number of data streams,
- $\mathbf{W}_{RF} \in \mathbb{C}^{N_{UE} \times N_{RF}}$ and $\mathbf{W}_{BB} \in \mathbb{C}^{N_{RF} \times N_s}$ are the analog and digital combiners at the UE side,
- $\boldsymbol{s} \in \mathbb{C}^{N_s \times 1}$ is the transmitted symbol vector,
- $\boldsymbol{n}(t) \sim \mathcal{CN}\left(0, \sigma_n^2 I\right)$ is the additive white Gaussian noise.

Equation (2) defines the hybrid beamforming system model, where the signal received at the UE is a combination of the transmitted signal through the analog precoding and digital precoding matrices at the BS as well as the analog and combining and digital combining matrices at the UE. Additive white Gaussian noise (AWGN) is assumed at the receiver with zero mean and variance σ_n^2 [13]. The system uses hybrid beamforming with separate analog and digital precoding/combining matrices to achieve power and phase control.

Figure 2 shows the concept of Doppler effect with an object moving at relative velocity v and the path angle is defined as θ between the source and direction of motion of the object.

The Doppler effect causes channel matrix $\mathbf{H}$ to vary over time. We model such a channel using a clustered geometric model, such as the Saleh-Valenzuela (SV) model [14] (Fig. 3). The channel matrix $\mathbf{H}(t)$ at time t is a superposition of the L scattering paths:

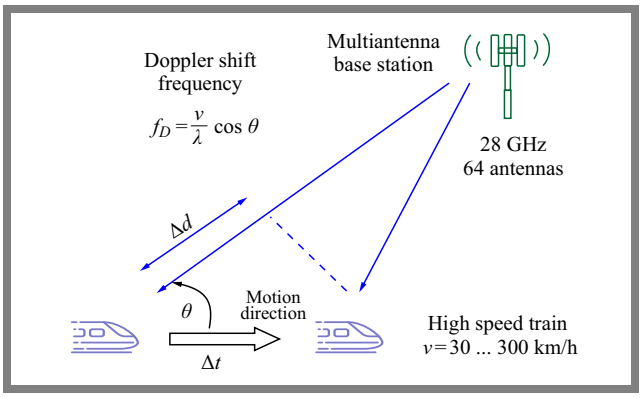

Fig. 2. Relative velocity v, path angle θ and Doppler shift.

$$\mathbf{H}(t) = \sqrt{\frac{N_t N_r}{L}} \sum_{l=1}^{L} \alpha_l e^{j2\pi f_{D,l} t} \boldsymbol{a}_{N_r}\left(\theta_l^{AOA}\right) \boldsymbol{a}_{N_t}^H\left(\theta_l^{AOD}\right), \tag{3}$$

where L is the number of paths, α_l is the complex gain of the l-th path, $f_{D,l} = \frac{v}{\lambda} \cos\left(\theta_l\right)$ is the Doppler frequency of l-th path, v is the user velocity, λ is the wavelength, and $\boldsymbol{a}(.)$ are the array response vectors.

This is the clustered geometric channel model, commonly used for mmWave channels, especially in high-mobility scenarios. Each path is associated with a complex gain α_l, and the Doppler shift $f_{D,l}$ introduces a time-varying phase shift at the receiver. Vectors $\boldsymbol{a}_{N_r}(\theta_l^{AOA})$ and $\boldsymbol{a}_{N_t}^H(\theta_l^{AOD})$ are the array response vectors for the UE and BS, respectively, corresponding to the angles of arrival $\theta_{r,l}$ and departure $\theta_{t,l}$.

3.2. LSTM-based Channel Prediction

The core of the proposed method is an LSTM network that predicts the $\mathbf{H}(t)$ future state of the channel matrix. The LSTM is particularly suitable for this task due to the ability to model long-term dependence in sequential data, which is essential for predicting channel variations in wireless communication systems. The input to the LSTM at time step t is

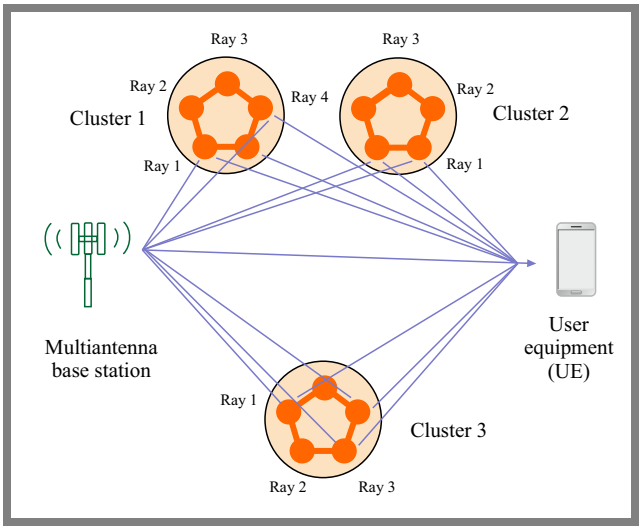

Fig. 3. Physical spread of the signal in the channel.

a sequence of the K most recent channel matrices:

$$X_t = \mathbf{H}(t-K), \mathbf{H}(t-K+1), \ldots, \mathbf{H}(t-1)\,,$$

where $\mathbf{H}(t) \in \mathbb{C}^{N_t \times N_r}$ is the channel matrix at time t, K is the number of previous time steps considered for predicting the channel matrix at the current time step.

These matrices are flattened and concatenated into a 2D tensor which serves as the input to the LSTM network. The LSTM network will use this sequence of channel matrices to predict the future state of the channel matrix.

We use a stacked LSTM architecture which involves multiple LSTM layers to capture both short- and long-term dependencies in the channel evolution:

- Input layer – takes the flattened channel sequence from the previous time steps.
- LSTM layer 1 – contains 128 LSTM units and processes the sequence and captures short-term temporal features. Using tanh as the activation function is standard.
- Layer 2 – with 64 LSTM units helps to learn more complex and longer-term dependencies in the evolution of the channel.
- Dense layer (fully connected) – with 256 neurons with ReLU activation function helps in mapping the learned temporal features to the desired output dimension.
- Output layer – a dense layer with $2 \times N_{UE} \times N_{BS}$ neurons (for real and imaginary parts) and a linear activation function to output the flattened predicted channel matrix $\widehat{\mathbf{H}}(t)$.

Each $\mathbf{H}(t) \in \mathbb{C}^{N_t \times N_r}$ is split into Re and Im and concatenated along the feature axis, yielding an input tensor of shape $(\mathrm{K}, 2N_rN_t)$.
The LSTM outputs $\widehat{\mathbf{H}}(\mathrm{t}+1)$ with the same format, which we reassemble into complex form. Beamformer update is realized using $\widehat{\mathbf{H}}(t+1)$ in following way:

- Compute SVD of: $H_{\text{pred}} = W_{RF}^H \widehat{\mathbf{H}}(\mathrm{t}+1)\mathbf{F}_{RF} \Rightarrow U \sum V^H$,
- Pick $\mathbf{F}_{RF}, \mathbf{W}_{RF}$ via OMP over steering-vector dictionaries to approximate the dominant singular subspace under constant-modulus,
- Set $\mathbf{F}_{BB} = V_{(:,1:N_S)}$ and $\mathbf{W}_{BB} = U_{(:,1:N_S)}$ with a normalization to meet power. This predict-then-design loop repeats every coherence block and K (history length) trades delay for robustness.

For network feature and I/O mapping each complex channel $\mathbf{H}(\mathrm{t}) \in \mathbb{C}^{N_t \times N_r}$ is split into real/imaginary parts and concatenated, yielding an input tensor shape $(\mathrm{K}, 2N_rN_t)$. We use a stacked LSTM with 128 and 64 hidden units (first returns sequences, second returns last state), followed by a dense 250 (ReLU) and an output layer of size $2N_rN_t$ (linear) that reconstructs $\widehat{\boldsymbol{H}}(t+\mathbf{1})\mathbb{C}^{N_r \times N_t}$. Then $\mathbf{H}_{eff} = \mathbf{W}_{RF}^H \widehat{\boldsymbol{H}}(t+1)\boldsymbol{F}_{RF}$ and design $\boldsymbol{F}_{BB}, \boldsymbol{W}_{BB}$ are formed via SVD subject to constant-modulus analog constraints as shown in Fig. 4.

The training process is conducted using the scheme presented below.

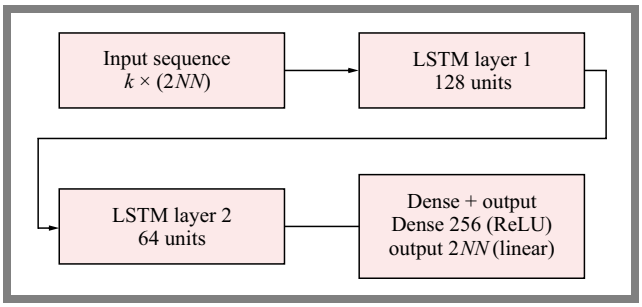

Fig. 4. LSTM network features.

We generate thousands of channel evolution sequences under various user velocities, angles, and scattering environments to create a comprehensive training dataset. This dataset represents the realistic dynamics of the channel matrix over time.

The loss function used to train the LSTM is the mean squared error (MSE) between the predicted channel matrix $\widehat{\mathbf{H}}(t)$ and the actual channel matrix $\mathbf{H}(t)$:

$$\mathcal{L} = \frac{1}{M} \sum_{i=1}^{M} \left\| \mathbf{H}_i(t) - \widehat{\mathbf{H}}_l(t) \right\|_F^2 , \tag{4}$$

where M is the batch size and $\|.\|_F$ is the Frobenius norm which calculates the matrix difference between the predicted and actual channel matrices.

The Adam optimizer is used for training the LSTM [15]. Adam is a popular optimization algorithm due to its adaptive learning rate and is well suited for training deep networks [16]. The LSTM-based predictive hybrid beamforming is provided as Algorithm 1, while Fig. 5 shows the flow of the predict-then-design hybrid beamforming method.

Algorithm 1 LSTM-based predictive hybrid beamforming

Input: Past K channels $\{\mathbf{H}(t-K+1), \ldots, \mathbf{H}(t)\} \in \mathbb{C}^{N_r \times N_t}$

1: **Complex → real stack:** $[\Re\{\mathbf{H}(.)\}, \Im\{\mathbf{H}(.)\}] \in \mathbb{R}^{K \times (2N_rN_t)}$
2: **LSTM prediction:** output $\widehat{\mathbf{H}}(t+1)\mathbb{C}^{N_r \times N_t}$
3: **Analog dictionaries:** $\mathcal{A}_t = \{a_{N_r}(\theta_m)\}, \mathcal{A}_r = \{a_{N_t}(\phi_n)\}$ (ULA, $d = \lambda/2$)
4: **OMP** (analog stage): select $\mathbf{F}_{RF} \in \mathbb{C}^{N_t \times N_{RF}^t}, \mathbf{W}_{RF} \in \mathbb{C}^{N_r \times N_{RF}^r}$ with $\left|[\mathbf{F}_{\mathrm{RF}}]_{ij}\right| = \frac{1}{\sqrt{N_t}}, \left|[\mathbf{W}_{\mathrm{RF}}]_{ij}\right| = \frac{1}{\sqrt{N_r}}$ to best approximate the dominant subspaces of $\widehat{\mathbf{H}}(t+1)$
5: **Digital stage:** $\mathbf{H}_{eff} = \mathbf{W}_{RF}^H \widehat{\mathbf{H}}(t+1)\mathbf{F}_{RF}$ take SVD $= U\Sigma V^H$; $\mathbf{F}_{BB} = V_{(:,1:N_s)}$ and $\mathbf{W}_{BB} = U_{(:,1:N_s)}$
Normalize $\|\mathbf{F}_{RF}\mathbf{F}_{BB}\|_F^2 = N_s$
6: **Apply:** use $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$ at slot $t+1$

Complexity: OMP $\mathcal{O}\left(N_s\left(|\mathcal{A}_t|\, N_t + |\mathcal{A}_r|\, N_r\right)\right)$;
SVD of $N_{RF}^r \times N_{RF}^t$
LSTM inference $\mathcal{O}\left(K, N_rN_td_{LSTM}\right)$

3.3. Predictive Hybrid Beamforming

The objective of predictive hybrid beamforming is to use predicted future channel states $\widehat{\mathbf{H}}(t)$ in order to adjust the beamforming matrices in a proactive manner, thus aiming to maximize spectral efficiency. This approach is especially beneficial in dynamic environments where the channel evolves

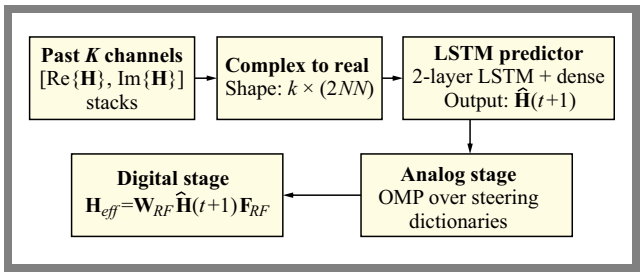

Fig. 5. Predict-then-design pipeline.

over time, such as high-mobility scenarios with significant Doppler shifts. By predicting the future state of the channel matrix, we can preemptively adjust the beamforming vectors to better align with the expected future channel conditions, improving the system performance.

In hybrid beamforming, the goal is typically to maximize the spectral efficiency, which quantifies the data rate achievable over a given bandwidth.

Let us define the effective channel as:

$$\mathbf{H}_{eff}(t) = \mathbf{W}_{RF}^{H}\,\mathbf{H}(t)\,\mathbf{F}_{RF}.$$

With digital processing, the per-slot spectral efficiency is:

$$R(t) = \log_2 \det\left(N_s + \frac{\rho}{N_s}\left(\mathbf{W}_{BB}^{H}\,\mathbf{W}_{BB}\right)^{-1}\cdot \mathbf{W}_{BB}^{H}\,\mathbf{H}_{eff}(t)\,\mathbf{F}_{BB}\,\mathbf{F}_{BB}^{H}\,\mathbf{H}_{eff}^{H}(t)\,\mathbf{W}_{BB}\right), \quad (5)$$

where $R(t)$ is the spectral efficiency in bits/sec/Hz and N_s is the number of data streams and $\rho = \frac{P}{\sigma_n^2}$.

In Eq. (5) the expression inside the logarithm is the effective channel capacity of the system, taking into account both transmit power and noise. The determinant represents the total capacity available in the system, considering the effective channel gain, interference, and noise.

The formulation assumes AWGN at the receiver and perfect channel state information at the transmitter. Term $\frac{P}{N_s\,\sigma_n^2}$ is the SNR per data stream. The determinant and logarithmic form comes from the Shannon capacity formula for MIMO systems, where the capacity grows logarithmically with the SNR, and the determinant represents the overall gain from the eigenvalues of the system which are captured by the SVD of the channel matrix.

We define $\mathbf{F}_{RF}, \mathbf{F}_{BB}, \mathbf{W}_{RF}, \mathbf{W}_{BB}$ using the predicted $\widehat{\mathbf{H}}(t+1)$ from the LSTM, subject to constant-modulus constraints on $\mathbf{F}_{RF}, \mathbf{W}_{RF}$ and a transmit-power constraint $||\mathbf{F}_{RF}\,\mathbf{F}_{BB}||_F^2 = N_s$.

The optimization problem is non-convex due to the constant-modulus constraint on the analog beamforming matrices [17], [18]. The constant-modulus constraint arises because the analog beamforming matrix (implemented using phase shifters) can only adjust the phase of each antenna element, but cannot control the amplitude.

The iterative approach is as follows:

- The first step is to design analog precoder $\mathbf{F}_{RF}$ and analog combiner $\mathbf{W}_{RF}$ to align with the dominant channel paths. This is achieved by selecting the columns from array response matrix $\alpha(\phi)$ from Eq. (1), which describes the response of the antenna array to different angles that maximize the projected channel gain.
- Once the analog beamformers $\mathbf{F}_{RF}$ and $\mathbf{W}_{RF}$ are fixed, the effective channel matrix at the receiver is:

$$\widehat{\mathbf{H}}_{eff} = \mathbf{W}_{RF}^{H}\widehat{\mathbf{H}}\,\mathbf{F}_{RF}\,. \quad (6)$$

This effective channel matrix represents the combined effect of analog beam formation at both the transmitter and the receiver.

Next, we develop the digital precoder $\mathbf{F}_{BB}$ and combiner $\mathbf{W}_{BB}$ by performing singular value decomposition (SVD) on the effective channel $\widehat{\mathbf{H}}_{eff}$:

$$\widehat{\mathbf{H}}_{eff} = \mathbf{U}_{eff}\,\Sigma\,\mathbf{V}^{\mathrm{H}}{}_{eff}\,\Sigma\,. \quad (7)$$

where $\mathbf{U}_{eff}$ and $\mathbf{V}_{eff}$ are the left and right singular matrices $\Sigma = \mathrm{diag}\,(\sigma_1, \ldots, \sigma_{N_S})$.

The digital precoder $\mathbf{F}_{BB}$ and digital combiner $\mathbf{W}_{BB}$ are derived from the right and left singular vectors of the effective channel:

$$\mathbf{F}_{BB} = \mathbf{V}_{eff}\,, \quad (8)$$

$$\mathbf{W}_{BB} = \mathbf{U}_{eff}\,. \quad (9)$$

These digital matrices align the signal with the strongest eigen modes of the effective channel and ensure optimal data stream transmission.

The key advantage of the predictive hybrid beamforming approach is that the beamforming matrices $\mathbf{F}_{RF}$ and $\mathbf{W}_{RF}$ are adjusted proactively based on the predicted future channel matrix $\widehat{\mathbf{H}}(t)$. This allows the system to track the mobile user effectively even in the presence of significant Doppler shifts.

By predicting the future state of the channel, the system can pre-emptively compensate for the variations caused by mobility, leading to more robust communication in high-mobility environments.

Per slot LSTM inference is $\mathcal{O}\,(\boldsymbol{K}, N_r\,N_t\,\boldsymbol{d}_{LSTM})$. OMP on steering dictionaries scales as $\mathcal{O}\,(N_s\,(|\mathcal{A}_t|\,N_t + |\mathcal{A}_r|\,N_r))$. The digital SVD runs on $\mathbf{H}_{eff} \in \mathbb{C}^{N_{RF}^r \times N_{RF}^t}$ (RF chain domain), costing $\mathcal{O}\big(\min\big\{(N_{RF}^r)^2\,N_{RF}^t, (N_{RF}^t)^2\,N_{RF}^r\big\}\big)$ not on full $N_r \times N_t$. Unlike full-digital beamforming, no large matrix inversion at antenna dimension is required, power normalization uses small $N_S \times N_S$ matrices.

4. Results and Discussion

Simulations are carried out in Matlab. Table 2 shows the simulation parameters and their values used to generate the results and discuss the concepts presented in this section.

The baseline models used for comparison are as follows.

- Orthogonal matching pursuit (OMP) based hybrid beamforming – a well-known iterative algorithm for hybrid precoding that does not account for Doppler effects [13],

Tab. 2. Simulation setup parameters.

Parameter	Value
Transmitting antenna N_t	64, tunable from 16 to 128
Receiving antenna N_r	16, tunable from 4 to 4
Transmitter RF chains N_{RF}^t	4, tunable from 2 to 8
Receiver RF chains N_{RF}^r	4, tunable from 2 to 8
Data streams N_s	4
Carrier frequency	28 GHz
Antenna spacing	$\frac{\lambda}{2}$
Number of channel paths	5
Mobility	Standardized mobility model (e.g. 3GPP urban mobility) 30, 120, 300 km/h

- Kalman filter-based hybrid beamforming (KF-HBF) – a conventional approach for tracking time-varying channels [7].
- Perfect CSI (upper bound) – an ideal case where instantaneous CSI is perfectly known at the transmitter [19].

We consider a mmWave system operating at 28 GHz with a 100 MHz bandwidth. The BS has a 64-element uniform linear array (ULA), and the UE has a 16- and 64-element ULA. We simulate the user velocities from 30 km/h to 300 km/h for different Doppler spreads.

Figure 6 shows the spectral efficiency (SE) plot as a function of SNR. At low SNR (–10 dB), all methods exhibit nearly equal performance (~ 4.5 bits/s/Hz). This convergence is expected due to noise dominance and limited beamforming gain. As SNR increases, divergence in performance becomes visible. At 0 dB, LSTM-HBF at 30 km/h achieves approximately 8.1 bits/s/Hz, which is slightly higher than KF-HBF (~ 7.9 bits/s/Hz) and OMP (~ 7.8 bits/s/Hz).

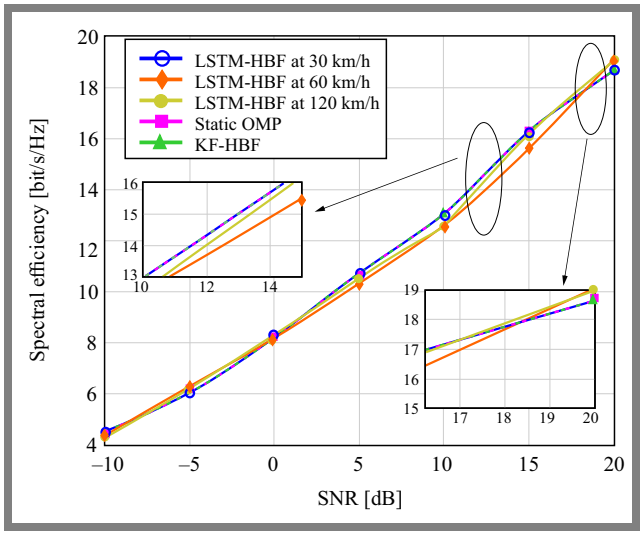

Fig. 6. Comparison of spectral efficiency for 64T and the 16R configuration and velocity.

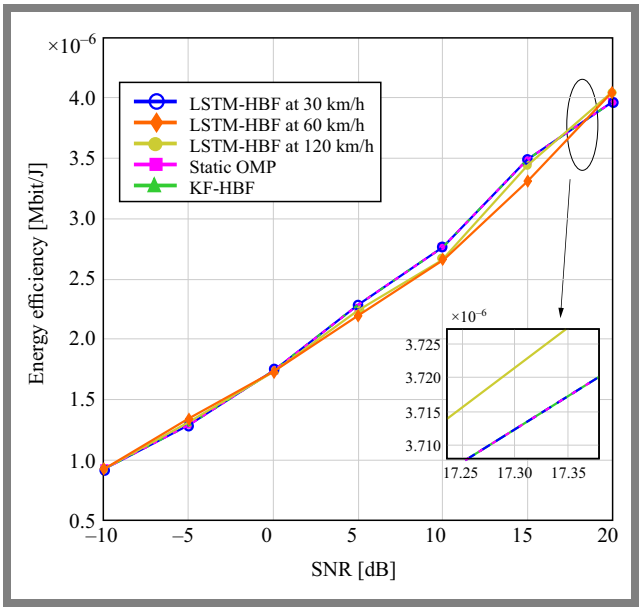

Fig. 7. Comparison of energy efficiency for 64T and 16R configuration and velocity.

At 10 dB, LSTM-HBF reaches ~ 13.7 bits/s/Hz, outperforming OMP and KF-HBF by ~ 0.5 bits/s/Hz, with a consistent margin across mobility variations. At 20 dB, LSTM-HBF peaks at approximately 19.2 bits/s/Hz, while OMP and KF-HBF saturate closer to 18.6 bits/s/Hz, suggesting that LSTM-based learning preserves marginal advantages even under high-SNR ceilings. Classical OMP/KF beamforming indeed performs well at high SNRs with low mobility. Our experiments target high-mobility Doppler where analog beams lag; the predictive design preserves alignment, yielding non-trivial throughput gains even when baselines appear near-optimal at 20 dB. This is consistent with the coherence-time limits at mmWave and supports proactive design rather than purely reactive tracking.

All methods are evaluated on identical channel realizations with identical power normalization. The mild divergence near 15 dB is the transition region from noise-limited to interference/quantization-limited operation, where analog dictionary granularity and prediction error interact, producing slightly different slopes across various methods.

Figure 7 shows energy efficiency (EE) as a function of SNR. At –10 dB SNR, all models operate near 0.9 Mbits/J, with LSTM-HBF at 60 km/h being slightly lower due to marginal underperformance in SE. From 0 dB onward, the EE curve shows more separation, at 5 dB, LSTM-HBF at 30 km/h achieves ~ 2.3 Mbits/J, leading OMP and KF-HBF by approximately 0.1 – 0.2 Mbits/J.

At 10 dB, LSTM-HBF continues its linear climb to ~ 3.4 Mbits/J, surpassing traditional methods by ~ 0.3 Mbits/J. By 20 dB, the advantage of EE becomes more noticeable with LSTM-HBF reaching ~ 4.2 Mbits/J, compared to OMP (~ 4.0) and KF-HBF (~ 4.05). The EE of LSTM-HBF at 120 km/h closely matches the results at 30 and 60 km/h, which is a strong indicator of model generalization under different Doppler conditions. KF-HBF shows slight degradation at

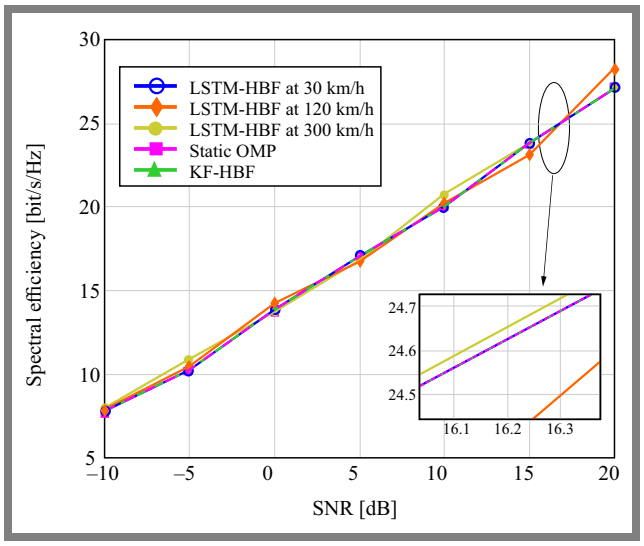

Fig. 8. Comparison of spectral efficiency for 64T and 64R configuration and velocity.

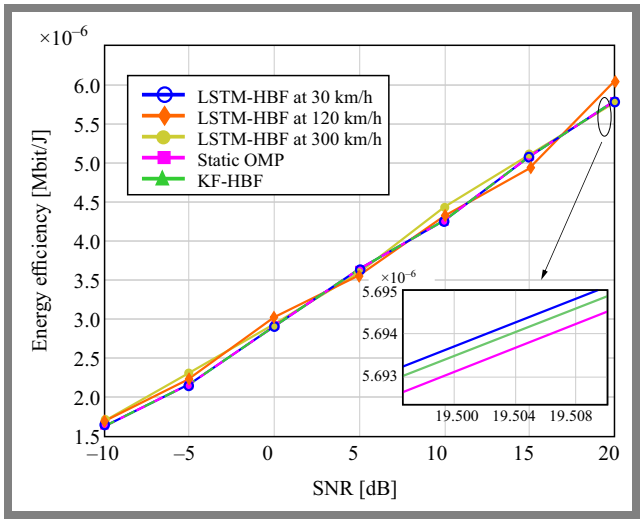

Fig. 9. Comparison of energy efficiency for 64T and 64R configuration and velocity.

higher SNRs, likely due to estimation errors accumulating over time, especially under high mobility.

Figure 8 illustrates the variation of spectral efficiency (SE) with respect to SNR for the proposed LSTM-HBF model at three different vehicular speeds: 30 km/h, 120 km/h, and 300 km/h, and compares it with two baseline methods: KF-HBF and static OMP. At an SNR of –10 dB, all schemes exhibit nearly identical SE, approximately 7.8 bits/s/Hz, as the system performance is primarily noise-limited. However, as the SNR increases, the LSTM-HBF method begins to demonstrate superior performance. At 0 dB, LSTM-HBF operating at 30 km/h achieves approximately 13.7 bits/s/Hz, slightly outperforming KF-HBF and OMP, which reach 13.3 and 13.2 bits/s/Hz, respectively. The performance gap becomes more pronounced at higher SNR values. At 20 dB, the LSTM-HBF model at 120 km/h attains a spectral efficiency of approximately 28.6 bits/s/Hz, outperforming KF-HBF and static OMP by roughly 1.1 and 1.3 bits/s/Hz, respectively.

Figure 9 presents the energy efficiency results for the same configurations. At an SNR of –10 dB, all models achieve comparable EE values of ∼1.6 to 1.7 Mbits/J. This similarity is expected given the low throughput and high relative power cost in this regime. As SNR increases, the LSTM-HBF model exhibits a more rapid improvement in EE. At 10 dB, the LSTM-HBF at 300 km/h achieves an EE of approximately 4.2×10^{-6} Mbits/J, slightly ahead of KF-HBF and OMP, which remain around 3.9×10^{-6} and 3.8×10^{-6} Mbits/J, respectively. By 20 dB, the LSTM-HBF model at 120 km/h reaches the highest energy efficiency of approximately 6.1×10^{-6} Mbits/J. In contrast, KF-HBF and static OMP attain 5.9×10^{-6} and 5.8×10^{-6} Mbits/J, respectively. These results highlight the dual advantage of LSTM-HBF in maximizing both spectral and energy efficiency.

5. Conclusions

This paper presents an LSTM-based hybrid beamforming (LSTM-HBF) approach that aims to improve performance of mmWave communication under varying mobility and SNR conditions. The method was benchmarked against traditional approaches such as static OMP and KF-HBF across various spectral and energy efficiency metrics.

The results demonstrate that LSTM-HBF consistently achieves higher performance. Especially, at 20 dB SNR, it delivers a spectral efficiency of up to 28.6 bits/s/Hz, compared to 27.5 bits/s/Hz for KF-HBF and 27.3 bits/s/Hz for static OMP. In terms of energy efficiency, LSTM HBF reaches 6.1×10^{-6} Mbits/J, surpassing the closest benchmark by a margin of $\sim 0.3 \times 10^{-6}$ Mbits/J. The LSTM-HBF model exhibits robust performance across all mobility profiles, spanning from 30 to 300 km/h, with minimal deviation in both SE and EE, indicating strong resilience to Doppler effects and time-varying channel conditions. Future work could explore more complex neural network architectures and investigate the impact of imperfect channel estimation on the performance of the proposed model.

By integrating a predictive LSTM network into a hybrid beamforming framework, we have shown through detailed modeling and simulated results that significant gains in both spectral and energy efficiency are achievable, particularly in high-mobility scenarios where traditional methods fail. This proactive approach ensures a robust communication link, making reliable multi-gigabit mobile communication a practical reality.

The future of this work involves real-life signal analysis with available datasets by extracting temporal channel slices along user trajectories and by using the same analog/digital design steps to compute SE/EE with identical power normalization.

References

[1] M. Shafi *et al.*, "5G: A Tutorial Overview of Standards, Trials, Challenges, Deployment, and Practice", *IEEE Journal on Selected Areas in Communications*, vol. 35, pp. 1201–1221, 2017 (https://doi.org/10.1109/JSAC.2017.2692307).

[2] T.S. Rappaport *et al.*, "Wireless Communications and Applications Above 100 GHz: Opportunities and Challenges for 6G and Beyond", *IEEE Access*, vol. 7, pp. 78729–78757, 2019 (https://doi.org/10.1109/ACCESS.2019.2921522).

[3] R.W. Heath *et al.*, "An Overview of Signal Processing Techniques for Millimeter Wave MIMO Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, pp. 436–453, 2016 (https://doi.org/10.1109/JSTSP.2016.2523924).

[4] A.F. Molisch *et al.*, "Hybrid Beamforming for Massive MIMO: A Survey", *IEEE Communications Magazine*, vol. 55, pp. 134–141, 2017 (https://doi.org/10.1109/MCOM.2017.1600400).

[5] M. Giordani, A. Zanella, and M. Zorzi, "Millimeter Wave Communication in Vehicular Networks: Challenges and Opportunities", *2017 6th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, Thessaloniki, Greece, 2017 (https://doi.org/10.1109/MOCAST.2017.7937682).

[6] A. Alkhateeb *et al.*, "Deep Learning Coordinated Beamforming for Highly-mobile Millimeter Wave Systems", *IEEE Access*, vol. 6, pp. 37328–37348, 2018 (https://doi.org/10.1109/ACCESS.2018.2850226).

[7] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A Tutorial on Particle Filters for Online Nonlinear/non-Gaussian Bayesian Tracking", *IEEE Transactions on Signal Processing*, vol. 50, pp. 174–188, 2002 (https://doi.org/10.1109/78.978374).

[8] H. Ye, G.Y. Li, and B.-H. Juang, "Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems", *IEEE Wireless Communications Letters*, vol. 7, pp. 114–117, 2018 (https://doi.org/10.1109/LWC.2017.2757490).

[9] E. Tuna and A. Soysal, "LSTM and GRU Based Traffic Prediction Using Live Network Data", *2021 29th Signal Processing and Communications Applications Conference (SIU)*, Istanbul, Turkey, 2021 (https://doi.org/10.1109/SIU53274.2021.9478011).

[10] C. Jiang *et al.*, "Machine Learning Paradigms for Next-generation Wireless Networks", *IEEE Wireless Communications*, vol. 24, pp. 98–105, 2017 (https://doi.org/10.1109/MWC.2016.1500356WC).

[11] W. Shahjehan *et al.*, "A Review on Millimeter-wave Hybrid Beamforming for Wireless Intelligent Transport Systems", *Future Internet*, vol. 16, art. no. 337, 2024 (https://doi.org/10.3390/fi16090337).

[12] A. Alkhateeb, O. El Ayach, G. Leus, and R.W. Heath, "Channel Estimation and Hybrid Precoding for Millimeter Wave Cellular Systems", *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, pp. 831–846, 2014 (https://doi.org/10.1109/JSTSP.2014.2334278).

[13] O. El Ayach *et al.*, "Spatially Sparse Precoding in Millimeter Wave MIMO Systems", *IEEE Transactions on Wireless Communications*, vol. 13, pp. 1499–1513, 2014 (https://doi.org/10.1109/TWC.2014.011714.130846).

[14] I. Marinovic, I. Zanchi, and Z. Blazevic, "Estimation of Channel Parameters for 'Saleh-Valenzuela' Model Simulation", *2005 18th International Conference on Applied Electromagnetics and Communications*, Dubrovnik, Croatia, 2005 (https://doi.org/10.1109/ICECOM.2005.204926).

[15] Z. Gao, L. Dai, Z. Wang, and S. Chen, "Spatially Common Sparsity Based Adaptive Channel Estimation and Feedback for FDD Massive MIMO", *IEEE Transactions on Signal Processing*, vol. 63, pp. 6169–6183, 2015 (https://doi.org/10.1109/TSP.2015.2463260).

[16] Junyi Wang *et al.*, "Beam Codebook Based Beamforming Protocol for Multi-Gbps Millimeter-wave WPAN Systems", *IEEE Journal on Selected Areas in Communications*, vol. 27, pp. 1390–1399, 2009 (https://doi.org/10.1109/JSAC.2009.091009).

[17] T. O'Shea and J. Hoydis, "An Introduction to Deep Learning for the Physical Layer", *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 563–575, 2017 (https://doi.org/10.1109/TCCN.2017.2758370).

[18] T. Wang, C.-K. Wen, S. Jin, and G.Y. Li, "Deep Learning-based CSI Feedback Approach for Time-varying Massive MIMO Channels", *IEEE Wireless Communications Letters*, vol. 8, pp. 416–419, 2019 (https://doi.org/10.1109/LWC.2018.2874264).

[19] S.H. Lim, S. Kim, B. Shim, and J.W. Choi, "Deep Learning-based Beam Tracking for Millimeter-wave Communications Under Mobility", *IEEE Transactions on Communications*, vol. 69, pp. 7458–7469, 2021 (https://doi.org/10.1109/TCOMM.2021.3107526).

Kartik Ramesh Patel, M.E.
Dep. of Electronics and Telecommunication Engineering
https://orcid.org/0000-0001-8391-5537
E-mail: kartik@somaiya.edu
MCT's Rajiv Gandhi Institute of Technology, Mumbai, India
https://www.mctrgit.ac.in

Sanjay Dasrao Deshmukh, Ph.D.
Dep. of Electronics and Telecommunication Engineering
https://orcid.org/0009-0004-0382-9831
E-mail: sanjay.deshmukh@mctrgit.ac.in
MCT's Rajiv Gandhi Institute of Technology, Mumbai, India
https://www.mctrgit.ac.in

Information for Authors

Journal of Telecommunications and Information Technology (JTIT) is published quarterly since 2000. It comprises original contributions, dealing with a wide range of topics related to telecommunications and information technology. **All papers are subject to peer review.** Topics presented in the JTIT report primary and/or experimental research results, which advance the base of scientific and technological knowledge about telecommunications and information technology.

JTIT is dedicated to publishing research results which advance the level of current research or add to the understanding of problems related to modulation and signal design, wireless communications, optical communications and photonic systems, voice communications devices, image and signal processing, transmission systems, network architecture, coding and communication theory, as well as information technology.

We encourage submissions from a diverse range of authors from across all countries and backgrounds.

Manuscript

Latex files are preferred and Editorial Office provides a style to prepare the material along with the documentation. We also accept Microsoft Word and PDF files. A typical article is 10 pages long (approximately 6,000 words) and must include the following contents:

- Authors' names and affiliations in the following format:
  First name and surname (last name), academic title,
  Position held,
  ORCID number,
  E-mail address from the University's domain,
  Faculty and name of the University,
  Link to University website.
- Abstract (150-200 words). The abstract should contain statement of the problem, assumptions and methodology, results and conclusion or discussion on the importance of the results. Abstracts must not include mathematical expressions or bibliographic references.
- Keywords related to the content of the article. About four keywords or phrases in alphabetical order should be used, separated by commas.
- The content of the article in a typical structure, i.e.: introduction, related work, conducted research, conclusions, references.

Figures, Tables and Photos

Together with the article, please send files with graphics with the highest resolution available, 150 dpi or more in bitmap resolution (jpg, png) and vector (cdr, svg, ps, pdf) formats are welcomed.

References

We use four main citation styles for a journal article, for an Internet article, for a conference paper, and for a book. Below are examples of citations. In each item, the DOI number or link to the PDF of the cited article should be provided.

[1] R.K. Meyers and A.H. Desoky, "An implementation of the blowfish cryptosystem", *2008 IEEE International Symposium on Signal Processing and Information Technology*, 2008 ((https://doi.org/10.1109/ISSPIT.2008.4775664).

[2] K. Nowicki and T. Uhl, *Ethernet End-to-End*, 1st ed. Germany, Shaker-Publisher, 2008 (ISBN: 978383832271404).

[3] C. Shorten and T.M. Khoshgoftaar, "A survey on image data augmentation for deep learning", *Journal of Big Data*, vol. 6, no. 1, pp. 1–48, 2019 (https://doi.org/10.1186/s40537-019-0197-0).

[4] S. Wong *et al.*, "Traffic forecasting using vehicle-to-vehicle communication", *3rd Annual Conference on Learning for Dynamics and Control*, pp. 917–929, 2021 (https://arxiv.org/pdf/2104.05528).

Submission

The paper with full PDF version and anonymous PDF version for the blind review process should be submitted on the JTIT website https://www.jtit.pl/jtit/about/submissions.

Reviewing Process

The article is initially approved by the Editor-In-Chief and if the decision is positive, is then sent to the reviewers. Depending on the subject of the article, it takes few weeks. In the next step, reviews are showed to authors who have 2 weeks to correct the article. Finally, the corrected text can be re-presented to the reviewer for reevaluation, which will take another 2 weeks.

As a result, after about 3 months, we are able to send the text for publication in the upcoming issue of JTIT.

When the reviews are inconsistent, additional corrections are necessary, or the reviewer expects additional verification because the corrections ordered by the author are insufficient or additional problems arise, the review of the article may be extended by another month or more.

Editorial Work

Positively reviewed and corrected article is next prepared by the editorial office for publication. At the end of this process the author receives an copyedited version for approval.

Licensing

Manuscript submitted to JTIT should not be published or simultaneously submitted for publication elsewhere. By submitting a manuscript the author grants license to the National Institute of Telecommunications, for the use of the paper in the fields of exploitation: reproducing and fixing the paper, distributing the paper by means of introduction to trade, letting for use or rental of the original or copies, and distributing the paper by means of public exhibition, screening, presentation and broadcast as well as rebroadcast, and making the paper publicly available in such a manner that anyone could access it at a place and time selected thereby, or by making it available in a way not allowing selection of time or place, including by means of Internet or other networks.

Ghostwriting Declaration

We require formal declaration that the process of writing the paper was not influenced by any third party. In the article, all the contributions of other people are clearly indicated. The theories presented, methods used, analysis and research, as well as the copyrights to the drawings, photographs and other figures belong to the authors or are clearly credited in the text. The author must also indicate whether his work has received financial support and if the realization of the whole project was possible thanks to the permission and cooperation with scientific institutions, associations and others.

Other Information

- The JTIT being an Open Access Journal (OAJ) has no article processing charges (APCs). The published articles can be downloaded freely without payment.
- JTIT supports open access and using continuous publishing "publish-as-you-go" scheme. This means that we no longer wait to accumulate several articles into a quarterly issue before publication. Rather, articles are continuously added to current issues after acceptation. Publish-as-you-go reduces publication lag for our authors, and make the newest research available quickly. After completing the review process, an article is published online in the current issue with DOI registration. When the issue period ends, a new issue is activated. So accepted articles are published without waiting for the quarterly issue end.

National Institute of Telecommunications

Editorial Office

National Institute
of Telecommunications
Szachowa st 1
04-894 Warsaw, Poland
https://www.gov.pl/web/instytut-lacznosci

phone +48 22 512 81 83
fax +48 22 512 84 00

e-mail: journal@jtit.pl
www.jtit.pl

JOURNAL OF TELECOMMUNICATIONS AND INFORMATION TECHNOLOGY 4/2025 vol.102