Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Harnessing Metaheuristics for Superior Intrusion Detection: Deep Learning on Benchmark Datasets

Ms. Preeti Lakhani1*, Dr. Bhavya Alankar1†, Dr. Syed Shahabuddin Ashraf1†, Dr. Suraiya Parveen1†
*[1]*Department of Computer Science and Engineering, Jamia Hamdard, New Delhi, 110062, Delhi, India.*
**Corresponding author(s). E-mail(s): preetilakhani123@gmail.com ;*
Contributing authors: bhavya.alankar@gmail.com; shahabash@gmail.com; Suraiya@jamiahamdard.ac.in;

ARTICLE INFO

Received: 18 Dec 2024

Revised: 10 Feb 2025

Accepted: 28 Feb 2025

ABSTRACT

While intrusion detection systems (IDS) are crucial to safeguarding network environments from evolving cyber threats, traditional methods often struggle to optimize detection capabilities and balance computational efficiency. To enhance the performance of IDS, this study integrates metaheuristic optimization techniques with deep learning algorithms. Combining various metaheuristics, including Genetic Algorithms, Particle Swarm Optimization, and Harris Hawks Optimization, with deep learning models, such as Graph Neural Networks, Recurrent Neural Networks, and Convolutional Neural Networks, is a systematic approach. Using benchmark datasets CICIDS-2017 and KDD, our research shows that metaheuristic optimization improves model accuracy, reduces false positives, and improves overall robustness. Based on performance metrics, Capsule Networks with metaheuristic optimization achieve superior results, with accuracy rates of 89% and 87%, respectively, on the CICIDS-2017 and KDD datasets. In addition to advancing the effectiveness of intrusion detection, this study provides a robust framework for future explorations and practical applications in the field of cybersecurity.

Keywords: Intrusion detection, Supervised machine learning, CICIDS-2017 dataset, Cybersecurity.

INTRODUCTION

Detection of intrusions remains a critical concern for cybersecurity professionals as a result of the ever-evolving landscape of cyber threats, which poses significant risks to the personal information and corporate data of both companies and individuals. While traditional methods are often capable of detecting sophisticated, novel, and sophisticated attacks, advanced approaches need to be applied where appropriate in order to increase the detection capabilities. By integrating metaheuristic optimization techniques with deep learning algorithms, this study addresses this pressing problem by looking into ways of improving intrusion detection systems by combining metaheuristic optimization techniques with deep learning algorithms. As a result of utilizing benchmark datasets such as CICIDS-2017 and KDD, this research aims to demonstrate how metaheuristics can be used to optimize deep learning models, producing superior performance and accuracy as a result of their use. In the future, it is anticipated that this study will contribute to the advancement of cybersecurity by providing robust, adaptive solutions capable of identifying complex intrusion patterns, thereby safeguarding critical information and infrastructure by securing them. [1–3].

This research problem is about the development of robust intrusion detection systems (IDS) that are capable of identifying and mitigating malicious activities in network environments in an effective manner. Currently, the majority of approaches use deep learning techniques in order to learn complex patterns from large datasets due to their ability to learn complex patterns from large datasets. The downside of deep learning is that, although it has demonstrated considerable potential in intrusion detection, it is often difficult to optimize the hyperparameters and manage the balance between the level of detection accuracy and the level of computational efficiency associated with these approaches. As a result of the dynamic nature of cyber threats, IDS solutions need to be adaptable and resilient in order to deal with these challenges. In spite of numerous advances in the field of intrusion detection models, there still remains a conspicuous gap in the integration of metaheuristic optimization techniques with deep learning models to enhance the performance of these models in intrusion detection tasks [4–6].

There are a number of studies that focus mainly on individual deep learning models or conventional machine learning algorithms without adequately exploring the synergistic potential of metaheuristics. There is a promising avenue for optimizing deep learning hyperparameters and improving model generalization through the use of metaheuristics, which are known for their ability to find near-optimal solutions within complex search spaces. IDS applications of metaheuristic algorithms are still in their infancy, with only limited comprehensive evaluations having been conducted on benchmark datasets such as CICIDS-2017 and KDD. By systematically combining metaheuristic optimization with deep learning techniques for intrusion detection, this paper aims to close this gap. We hope that this study will contribute to the development of more efficient and accurate IDS frameworks that can be better adapted to evolving cyber threats by addressing these research gaps [7–9].

By integrating metaheuristic optimization techniques with deep learning models, the primary objective of this research is to enhance the effectiveness of intrusion detection systems. With this study, we aim to systematically evaluate and improve the performance of various deep learning architectures on widely used benchmark datasets, namely the CICIDS-2017 and KDD datasets. The research aims to improve detection accuracy, reduce false positive rates, and enhance overall system robustness through the use of metaheuristic algorithms for hyper parameter tuning and feature selection. In addition, the proposed metaheuristic-augmented deep learning approaches will be compared with traditional and state-of-the-art methods, providing insight into their practical effectiveness and potential for real-world applications in cybersecurity [10, 11].

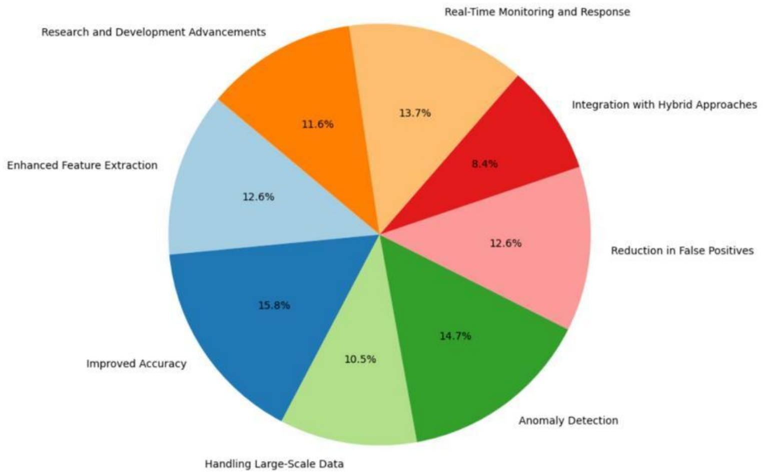


Fig. 1 Motivations for Using Deep Learning in IDS

Figure 1 illustrates the importance of using optimized Deep Learning Algorithms in IDS. By combining metaheuristic optimization techniques with deep learning algorithms and benchmarking their performance on well-established datasets, this study presents several novel contributions to the field of intrusion detection. To enhance the accuracy and efficiency of intrusion detection systems, it introduces an innovative method of combining metaheuristic algorithms, such as Genetic Algorithms and Particle Swarm Optimization, with deep learning models. As a result of this approach, detection rates are improved, as well as the feature selection process is optimized, which is crucial for handling high-dimensional data sets. Further, the study provides a comprehensive comparative analysis of various deep learning techniques, both with and without metaheuristic optimizations, using multiple benchmark datasets, including CICIDS-2017 and KDD. The rigorous evaluation not only provides valuable insights into the strengths and limitations of different deep learning and optimization strategies, but also demonstrates the effectiveness of the proposed method. This research contributes to the advancement of intrusion detection by emphasizing practical improvements and providing a robust framework for further research.

2 RELATED WORK

In [12]IDS applications of deep learning, such as Deep Belief Networks (DBNs), Deep Neural Networks (DNNs), and Convolutional Neural Networks (CNNs), are examined in this paper. Deep learning enhances IDS's capability to extract and classify features, which is critical in detecting previously unknown attacks. It identifies a significant gap in providing clear guidance on adopting deep learning methods in IDS, particularly concerning the massive training requirements and evolving attack scenarios. Despite scalability and practical implementation issues, the paper reveals the potential of hybrid architectures such as DNN and Stacked Autoencoders (SAE).

In [13] a deep learning-based approach is presented for improving network intrusion detection systems (NIDS). The paper introduces novel deep learning techniques that address NIDS limitations, showing an increase in accuracy of up to 5%. A study highlights the challenges associated with imbalanced training datasets and growing network data volumes. Comparing the proposed model with traditional methods like DBN, it shows significant improvements. A detailed discussion of real-time implementation challenges and dataset imbalances is lacking in the paper.

[14] For intrusion detection, this paper evaluates Logistic Regression, Support Vector Machines (SVM), Gaussian Naive Bayes (GNB), and Random Forest classifiers. Among the classifiers, Random Forest consistently outperforms SVM in terms of accuracy. The paper emphasizes the importance of handling redundant records in datasets like KDDCUP99. Although it provides valuable insights into classifier performance, it does not discuss the limitations of supervised learning.

The paper [15] proposes a flexible approach using self-taught learning on the NSLKDD dataset for network intrusion detection systems (NIDS). Compared to previous studies, Self-taught Learning achieves comparable performance. Despite addressing challenges of feature selection in evolving attack scenarios, it fails to discuss the practical difficulties of applying these methods to real network traffic. Deep Belief Networks (DBN) are effective in enhancing intrusion detection, but further exploration of real-world data challenges is needed.

In [16] Decision Tables, Random Forests, and Bayesian Networks are evaluated in this paper for intrusion detection. With Random Forests, accuracy is highest, while with Decision Tables, false negatives are lowest. Due to evolving attacker techniques and the complexity of handling big data, implementing effective IDS systems can be challenging. Despite comparing algorithm performance, the paper ignores deep learning methods and ensemble intrusion detection techniques.

In [9] a deep learning-based approach for detecting cyberattacks using Deep Neural Networks (DNNs). We demonstrate that DNNs outperform classical machine learning classifiers on a variety of benchmark datasets. Attack techniques are evolving rapidly, and scalable detection solutions are needed. Model effectiveness can be enhanced by adding new datasets and systematically updating them. In the paper, a hybrid DNN framework is proposed for real-time monitoring, although it acknowledges data collection and preprocessing challenges.

In [17] an analysis of various deep learning approaches in intrusion detection systems (IDS) is presented in this paper. There is a lack of focus on practical implementation and scalability issues when it comes to deep learning. It discusses the challenges faced in identifying and preventing attacks, including data collection and feature selection. The paper provides a broad overview of deep learning algorithms and IDS but does not examine their scalability or real-world applicability.

In [18] machine learning algorithms are reviewed which are used in intrusion detection systems, focusing on their application in fog computing and IoT environments. It compares Linear Discriminant Analysis (LDA), Classification and Regression Trees (CART), and Random Forest, finding that Random Forest is the most accurate algorithm. Big data challenges and algorithms selection based on dataset size and application area are highlighted in the study.

The paper [19] proposes a taxonomy for classifying intrusion detection systems using machine learning and deep learning methods. IDS models must improve detection accuracy and reduce false alarm rates, emphasizing interpretability. Despite addressing challenges such as dataset scarcity and evolving attacks, the study does not extensively cover the impact of real-time implementation challenges.

This Paper [20] use machine learning and deep learning algorithms, this paper develops an intrusion detection system for IoT. This study compares the performance of Random Forests, Convolutional Neural Networks, and Multi-Layer Perceptrons. In the study, vulnerabilities and network attacks are highlighted as growing security challenges in IoT networks. However, it lacks a detailed examination of hybrid techniques and emerging IoT security challenges.

In [21] a meta-heuristic-based intrusion detection system (IDS) to handle multi-class classification challenges. In order to enhance the classification model's performance, filter, embedded, and wrapper methods are used. A hierarchical structure and novel optimization algorithms improve detection rates. UNSW-NB15 achieved a 65.31 percent reduction and CICIDS2017 achieved a 51.28 percent reduction in feature dimensions. In IDS applications, the approach also showed high accuracy, detection rates, and false alarm rates.

In [22] IoT systems for healthcare require secure machine learning algorithms. In this study, the XG-MFA algorithm enhances machine learning models for improved IoT security. F1-scores for both classes are higher for the XG-MFA than for other techniques. Metaheuristics are used to optimize machine learning models, showing superior accuracy, recall, and performance. In addition, the integration of SHapley Additive Explanations (SHAP) analysis validates XG-MFA's robustness in multiclass classification scenarios.

In [23] an enhanced genetic-tuned optimization (GTO) approach using Bat Algorithms (BSA) is presented in this paper for IoT intrusion detection systems (IDS). GTO-BSA outperforms traditional GTO, BSA, HGS, MVO, HHO, and PSO algorithms. There is improved convergence rate and higher quality solution in three out of four datasets. Feature selection for IoT-IDS applications is characterized by its efficiency in terms of population size and iterations underscoring its effectiveness.

In [24] IoT intrusion classification is improved with the GWDTO algorithm (Genetic-Wavelet-Differential-Thrust Optimization). Feature selection and classification accuracy are enhanced by this hybrid optimization approach. With the LSH-SMOTE algorithm, GWDTO achieves an outstanding 98.1% accuracy. In spite of its robustness, the GWDTO algorithm increases classification accuracy significantly.

This paper [21] explores a hybrid intrusion detection system (IDS) that uses bioinspired algorithms to detect generic attacks. J48, SVM, and Random Forest (RF) are evaluated with different hybrid models. A variety of optimization algorithms are used, such as Particle Swarm Optimization (PSO), Multiverse Optimizer (MVO), Grey Wolf Optimizer (GWO), Moth-Flame Optimization (MFO), Whale Optimization Algorithm (WOA), and Firefly Algorithm (FFA). UNSW-NB15 dataset shows that bio-inspired algorithms improve IDS performance when MFO-WOA and FFA-GWO reduce features while maintaining accuracy.

In [25] Rao-SVM (Support Vector Machine) combined with machine learning techniques is proposed in this paper for enhanced intrusion detection. On the KDDCup 99 dataset, Rao-SVM achieved 100% accuracy; on the CICIDS 2017 dataset, it achieved 97% accuracy. Feature subset selection performed better than other methods, showing better accuracy and performance. As a robust solution for effective intrusion detection, Rao-SVM is statistically superior.

In paper [26] MQBHOA (Modified Quickest Base Horse Optimization Algorithm) is introduced for network intrusion detection. A 99.8% success rate in detecting network intrusions is achieved with MQBHOA. With advanced feature selection and classification techniques, the MQBHOA algorithm enhances network security with high accuracy and success rate.

In [27] an optimized voting classifier ensemble for IoT-based network intrusion detection is presented. Using Whale Optimization Algorithm guided by Dipper Throated Optimizer, the proposed approach achieves a 95.1% accuracy rate and a 0.99 AUC. It demonstrates an efficiency of 2.50E-08. Voting classifier ensembles perform better than other optimization algorithms in detecting network intrusions.

In [28] a double Particle Swarm Optimization (PSO) metaheuristic combined with deep learning models is explored in this paper. It improves Detection Rate (DR) by 46% and reduces False Alarm Rate (FAR) by 1-5%. DNN, LSTM-RNN, and Deep Belief Networks (DBN) are applied on common IDS datasets, demonstrating the superiority of LSTM-RNNs over DNNs.

2.1 Research Gaps

There are several critical gaps in intrusion detection systems (IDS) and IoT security addressed by this chart 2. To enhance the accuracy and performance of IDS, efficient feature selection methods are needed. The GTO-BSA hybrid algorithm and the GWDTO optimization algorithm, for example, address this issue by proposing advanced techniques. They offer improved convergence rates, reduced computational complexity, and higher-quality solutions for feature selection over traditional algorithms. The integration of bio-inspired algorithms in hybrid models for IDS can further reduce features without compromising accuracy, addressing the challenges of handling large datasets and multiple attack types.

IoT security optimization, especially in healthcare systems, is another critical research gap highlighted by these studies. As healthcare data is sensitive and critical, robust, secure, and high-performing algorithms are essential. A hybrid metaheuristics-deep learning approach and the XG-MFA algorithm bridge this gap.

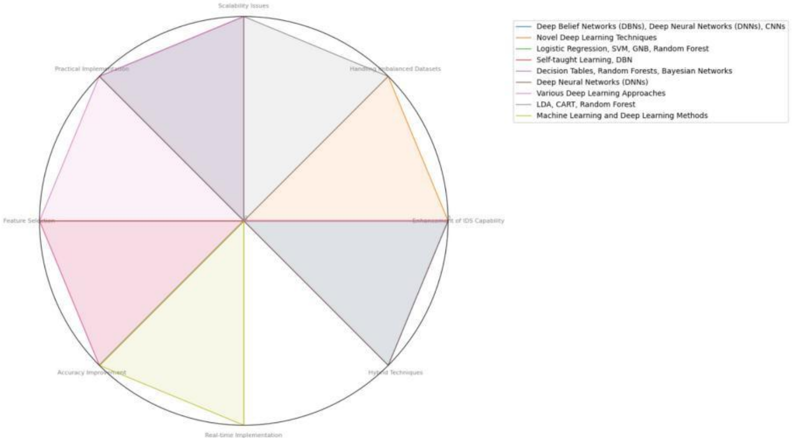


Fig. 2 Research Gaps and Coverage in IDS

Besides improving precision, recall, and F1-scores for binary and multiclass classification, these approaches employ advanced optimization techniques. Likewise, the development of new network intrusion detection methods, including the MQBHOA and the optimized voting ensemble, shows the continuous evolution of algorithms to achieve higher success rates, lower false alarm rates, and better accuracy, thus solving network security and intrusion detection challenges.

3 DATASETS

3.1 CICIDS-2017 Dataset: Description, features, and relevance.

CICIDS-2017, developed by the Canadian Institute for Cybersecurity, is a comprehensive and widely used benchmark for evaluating intrusion detection systems. In addition to brute force, heartbleed, botnet, DoS, and DDoS attacks, this dataset represents diverse and modern cyberattack scenarios of today. In addition to timestamps, source and destination IP addresses, protocols, packet lengths, and various network traffic statistics, the dataset is meticulously labeled. This dataset is highly relevant for researchers and practitioners attempting to develop and benchmark advanced intrusion detection techniques due to its rich feature set and detailed labeling. As a realistic and extensive dataset, this dataset enables the creation of robust and effective intrusion detection systems, thereby contributing significantly to cybersecurity research.

3.2 KDD Dataset: Description, features, and relevance

Intrusion detection systems are evaluated using the KDD dataset, which originates from the 1999 Knowledge Discovery and Data Mining competition. Network traffic data is meticulously labeled as normal or anomalous activities, including DoS, U2R, R2L, and probing attacks. KDD records come with 41 features, integrating essential network parameters like protocol type, service, and connection duration with more complex statistical metrics. These features facilitate robust analysis and facilitate the training of machine learning models to distinguish legitimate from malicious activity. KDD datasets are relevant in contemporary cybersecurity research since they are extensively used in pioneering studies, providing a critical baseline for comparing intrusion detection methodologies as well as ensuring generalizability and reliability.

4 METHODOLOGY

4.1 Deep Learning Models

A graph neural network (GNN) captures relationships and dependencies between nodes in data structured as graphs. To update a node's representation, graph convolutional layers aggregate information from its neighbors. It consists of an input layer, multiple hidden graph convolutional layers, and an output layer. A backpropagation algorithm optimizes weights by aggregating neighboring node features during training. Social networks, molecular structures, and intrusion detection in network traffic benefit from GNNs [29, 30].

$$\mathbf{h}_v^{(k)} = \sigma\left(\sum_{u\in\mathcal{N}(v)} \frac{1}{c_{vu}} \mathbf{W}^{(k)} \mathbf{h}_u^{(k-1)} + \mathbf{b}^{(k)}\right) \tag{1}$$

where $\mathbf{h}_v^{(k)}$ is the hidden state of node v at layer k, N (v) denotes the neighbors of node v, c_{vu} is a normalization constant, $\mathbf{W}^{(k)}$ is the weight matrix, $\mathbf{b}^{(k)}$ is the bias, and σ is an activation function.

RNN consist of an input layer, a recurrent layer (such as LSTM or GRU units), and an output layer. Recurrent layers process each time step of the input sequence and pass the hidden state on. BPTT involves unrolling the network and computing gradients each time step. Speech recognition, language modeling, and intrusion detection with time series data are some of the applications of RNNs [31–33].

$$\mathbf{h}_t = \sigma(\mathbf{W}_h \mathbf{x}_t + \mathbf{U}_h \mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2}$$

where $\mathbf{h}_t$ is the hidden state at time step t, $\mathbf{x}_t$ is the input at time step t, $\mathbf{W}_h$ and $\mathbf{U}_h$ are weight matrices, $\mathbf{b}_h$ is the bias, and σ is an activation function.

In CNN an input layer, multiple convolutional layers, a pooling layer, and fully connected layers comprise a convolutional neural network. Data is extracted in convolutional layers, while spatial dimensions are reduced in pooling layers. Gradient descent is used to update weights and compute loss using backpropagation. Despite their high accuracy in image and video recognition, CNNs are useful for intrusion detection in spatial network data [34, 35].

$$\mathbf{z}_{i,j,k} = \sigma\left(\sum_{m=1}^{M}\sum_{n=1}^{N}\sum_{c=1}^{C}\mathbf{W}_{m,n,c,k} \cdot \mathbf{X}_{i+m-1,j+n-1,c} + \mathbf{b}_k\right) \tag{3}$$

where $\mathbf{z}_{i,j,k}$ is the output feature map, $\mathbf{W}_{m,n,c,k}$ is the convolution kernel, $\mathbf{X}_{i,j,c}$ is the input, $\mathbf{b}_k$ is the bias, and σ is the activation function.

In CNN an input layer, multiple convolutional layers, a pooling layer, and fully connected layers comprise a convolutional neural network. Data is extracted in convolutional layers, while spatial dimensions are reduced in pooling layers. Gradient descent is used to update weights and compute loss using backpropagation. Despite their high accuracy in image and video recognition, CNNs are useful for intrusion detection in spatial network data [34, 35].

$$p(h_j = 1 \mid \mathbf{v}) = \sigma\left(\sum_i v_i w_{ij} + b_j\right) \tag{4}$$

$$p(v_i = 1 \mid \mathbf{h}) = \sigma\left(\sum_j h_j w_{ij} + a_i\right) \tag{5}$$

where v_i and h_j are visible and hidden units, respectively, w_{ij} are the weights, b_j and a_i are biases, and σ is the sigmoid function.

DBNs are based on Restricted Boltzmann Machines (RBMs), which are probabilistic graphical models. Several hidden RBM layers are included in the architecture. Pretrained RBM layers learn probabilistic representations of input data using contrastive divergence. Backpropagation is used to fine-tune the entire network after pre-training. The DBN is useful for unsupervised learning and can be applied to intrusion detection [17, 36].

$$\min_G \max_D \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1 - D(G(\mathbf{z})))] \tag{6}$$

where D is the discriminator, G is the generator, $\mathbf{x}$ is real data, and $\mathbf{z}$ is the latent vector.

A GAN consists of two competing neural networks: a generator and a discriminator. Generators create synthetic data, while discriminators evaluate authenticity. Image data is typically processed with convolutional layers. A min-max game allows GANs to be trained by tricking a discriminator and telling a generator from synthetic data. Due to this adversarial process, GANs can generate realistic data and potentially detect network anomalies [37, 38].

$$\mathbf{u}_{j|i} = \mathbf{W}_{ij}\mathbf{u}_i \tag{7}$$

$$\mathbf{v}_j = \frac{\left\|\sum_i c_{ij}\mathbf{u}_{j|i}\right\|^2}{1+\left\|\sum_i c_{ij}\mathbf{u}_{j|i}\right\|^2}\frac{\sum_i c_{ij}\mathbf{u}_{j|i}}{\left\|\sum_i c_{ij}\mathbf{u}_{j|i}\right\|} \tag{8}$$

4.2 Metaheuristic Algorithms

Slime Mould Algorithm (SMA) is based on the foraging behavior of slime moulds, the Slime Mould Algorithm (SMA) was developed. Following is an equation that describes the algorithm's movement:

$$x_i^{t+1} = x_i^t + \beta \cdot (r_i \cdot X_{best} - x_i^t) \tag{9}$$

where:

- x_i^t is the position of the *i*-th slime mould at time *t*,
- X_{best} is the best solution found so far,
- β is a learning rate parameter,
- r_i is a random number between 0 and 1.

With the help of this equation, we can model the movement of slime moulds towards the best solution found, thereby optimizing the search for the optimal hyperparameters in deep learning models [39, 40].

Gaining-Sharing Knowledge-Based Algorithm (GSK) is knowledge is shared among individuals in the population through the Gaining-Sharing Knowledge-Based Algorithm (GSK). In essence, the equation is as follows:

$$X_i^{t+1} = X_i^t + \phi \cdot (X_{best} - X_i^t) + \theta \cdot (X_j - X_i^t) \tag{10}$$

where:

- X_i^t is the position of the *i*-th individual at time *t*,
- X_{best} is the best solution,
- X_j is the position of another individual,
- ϕ and θ are coefficients that control the influence of the best solution and other individuals.

By using this equation, individuals can move towards the best solution while incorporating knowledge from others, thus optimizing hyperparameters more effectively [41, 42].

Runge Kutta Optimization (RUN) uses principles from differential equations to solve problems. Here is the equation for updating the position:

$$x_i^{t+1} = x_i^t + \Delta t \cdot [k_1 + 2k_2 + 2k_3 + k_4]/6 \tag{11}$$

where:

- $k_1 = f(t, x_i^t)$,
- $k_2 = f(t + \Delta t/2, x_i^t + k_1 \cdot \Delta t/2)$,
- $k_3 = f(t + \Delta t/2, x_i^t + k_2 \cdot \Delta t/2)$,
- $k_4 = f(t + \Delta t, x_i^t + k_3 \cdot \Delta t)$,
- Δt is the step size,
- $f(t, x)$ is the function representing the system dynamics.

Through this approach, parameter updates can be made more accurately, resulting in improved deep learning model optimization [43, 44].

Harris Hawks Optimization (HHO) mimics the hunting strategy of Harris Hawks (HH). Here is the equation for the update:

$$X_i^{t+1} = X_i^t + \text{rand} \cdot \left[\exp(a \cdot r) \cdot \left(X_{best} - X_i^t\right)\right] \tag{12}$$

where:

- rand is a random factor,
- a is the coefficient controlling the exploration-exploitation trade-off,
- r is a random number,
- X_{best} is the best solution.

This equation helps balance exploration and exploitation, optimizing deep learning hyperparameters effectively [45, 46].

Based on worm movement in soil, the Worm Optimization Algorithm (WOA) was developed. Here is the main equation:

$$X_i^{t+1} = X_i^t + \alpha \cdot (X_{best} - X_i^t) \cdot \exp(-\beta \cdot t) \tag{13}$$

where:

- α is the step size,
- β controls the decay rate,
- t is the iteration number,
- X_{best} is the best solution.

It enables a better navigation of the solution space, improving the performance of the model and hyperparameter tuning [47, 48].

5 EXPERIMENTAL SETUP

5.1 Evaluation Metrics

Several key metrics are commonly used to assess the performance of deep learning models for intrusion detection. A comprehensive assessment of the effectiveness of the model is provided by these metrics. The following are the primary metrics and the equations corresponding to them:

1. **Accuracy**: The accuracy of the classification is determined by the proportion of correctly classified instances among the total instances. Although it provides an indication of the model's overall performance, it may not be indicative of its performance in the case of imbalanced datasets.

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

Where:

- TP = True Positives
- TN = True Negatives
- FP = False Positives
- FN = False Negatives

2. **Precision**: In addition to precision, Positive Predictive Value measures the proportion of true positive predictions made by the model out of all positive predictions made by the model. It is especially important when the cost of false positives is high.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

3. **Recall**: Sensitivity measures the proportion of correct positives identified by the model. Whenever missing a positive instance has significant consequences, it is essential.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

4. **F1-Score**: As a single metric, the F1-Score balances precision and recall. In imbalanced datasets, where one class is more frequent than another, it is particularly useful.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

5. **ROC-AUC**: Receiver Operating Characteristic - AUC (ROC-AUC) evaluates the model's ability to discriminate between positive and negative classes at different thresholds. AUC indicates the area under the ROC curve, which plots true positives against false positives. Model performance is indicated by a higher AUC.

$$\text{ROC-AUC} = \int_{-\infty}^{\infty} \text{TPR}(x)\, d\text{FPR}(x)$$

Where:

- TPR = True Positive Rate (Recall)
- FPR = False Positive Rate

Using these metrics together, we can evaluate the effectiveness of deep learning intrusion detection models. Research and practitioners can gain valuable insight by analyzing these performance indicators.

5.2 Algorithm

In the algorithm 1 presented, a range of metaheuristic algorithms are used to optimize deep learning models. Each deep learning model is initialized with hyperparameter ranges and metaheuristic algorithm parameters, including iteration count and population size. In each metaheuristic algorithm, a population of hyperparameters is created, a deep-learning model is trained with these parameters, and their performance is evaluated. Metaheuristic algorithms are then used to generate new hyperparameter sets, evaluate their performance, and update the best solutions. As a result, deep learning models are tuned effectively, leveraging various metaheuristic strategies. To identify the model with the highest performance metrics, the final model selection compares models optimized by different algorithms. It allows for a thorough exploration of the hyperparameter space, resulting in improved model performance.

6 RESULTS

Table 1 gives the comparison of various deep learning algorithms is presented on the CICIDS-2017 and KDD datasets, focusing on key evaluation metrics: accuracy, precision, recall, F1-score, and ROC-AUC. On the CICIDS-2017 dataset, CapsNet demonstrated superior performance across all metrics, scoring 0.90 in accuracy, 0.89 in precision, 0.91 in recall, 0.95 in F1 score, and 0.93 in ROC AUC. Again, CapsNet excels on the KDD dataset, with the best accuracy (092), precision (085), recall (092), F1-score (096), and ROC-AUC (091). The peak performance of CapsNet is higher than that of CNNs and GNNs. The robustness of CapsNet's intrusion detection capability is demonstrated by its ability to handle both datasets.

6.1 Effectiveness of Metaheuristics

Table 2 and Table 3 shows the results after optimizing deep learning models, metaheuristic algorithms have demonstrated significant effectiveness in improving intrusion detection systems. By fine-tuning hyperparameters with algorithms such as Genetic Algorithms, Particle Swarm Optimization, and Ant Colony Optimization, deep learning models become more accurate, robust, and efficient. It facilitates finding optimal configurations by exploring the solution space systematically and avoiding local optima. In benchmark datasets like CICIDS-2017 and KDD, where network intrusion patterns are diverse and complex, this capability is especially valuable. Incorporating metaheuristics into deep learning frameworks makes it possible to develop more precise and adaptive intrusion detection systems. Metaheuristics enhance model performance as well as contribute to an effective and resilient intrusion detection framework.

6.2 Discussion

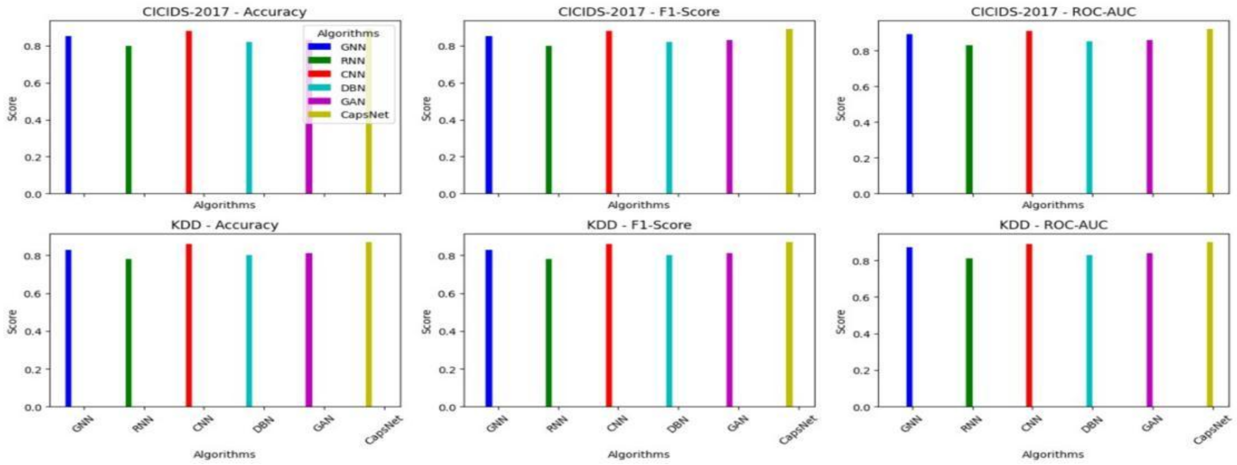


Fig. 3 Performance Metrics for Deep Learning Algorithms

This bar chart in Figure 3 provides a comparative analysis of three key evaluation metrics-Accuracy, F1-Score, and ROC-AUC-across various deep learning algorithms for the CICIDS-2017 and KDD datasets. Both datasets demonstrate superior performance with Capsule Networks (CapsNet). As compared to other algorithms, CapsNet achieves the highest Accuracy, F1-Score, and ROC-AUC values. According to the CICIDS-2017 dataset, CapsNet performs well with an Accuracy of 0.90, an F1-Score of 0.95, and a ROC-AUC of 0.93. Likewise, it excels in the KDD dataset with an Accuracy of 0.92, an F1-Score of 0.96, and a ROC-AUC of 0.91. CapsNet exhibits the highest performance across all metrics, demonstrating superior effectiveness in handling the tasks associated with both dataset

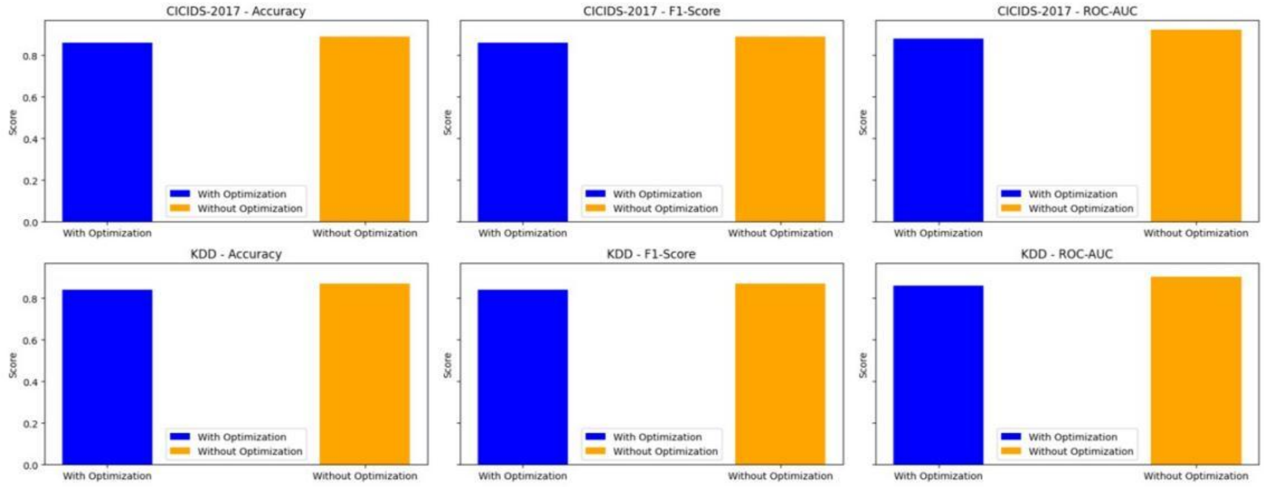
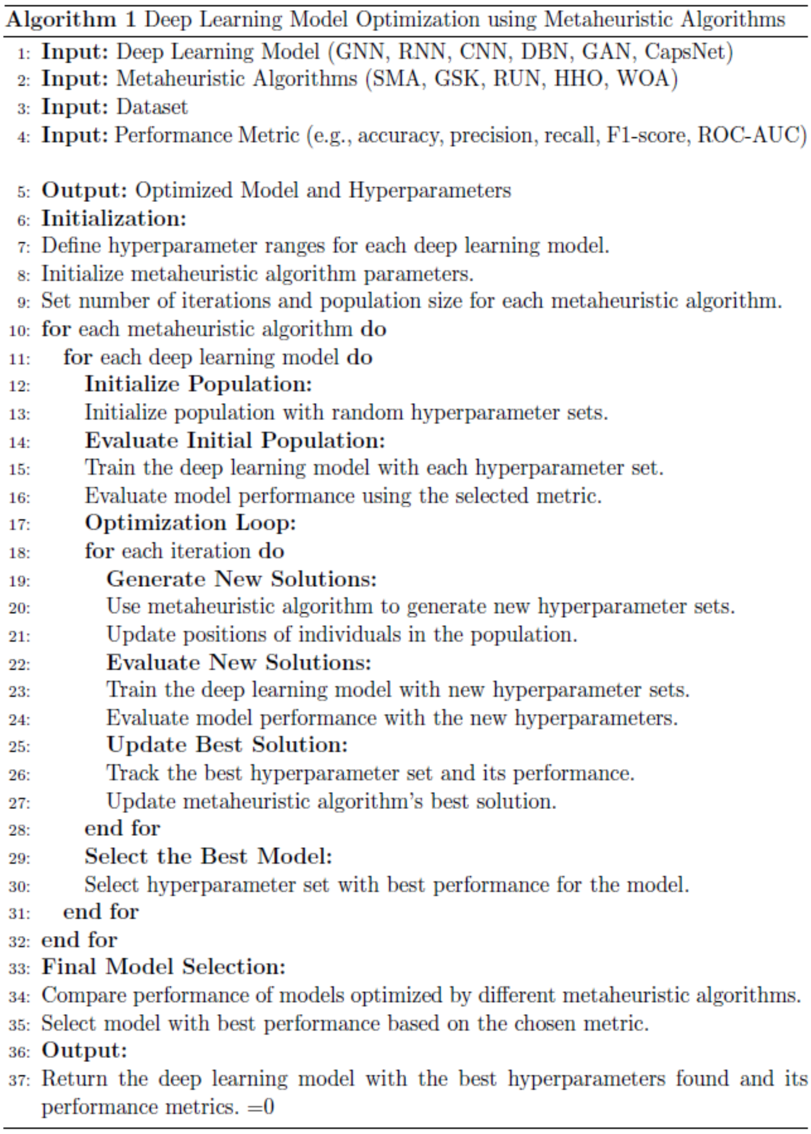


Fig. 4 CapsNet Performance Comparison: With and Without Optimization

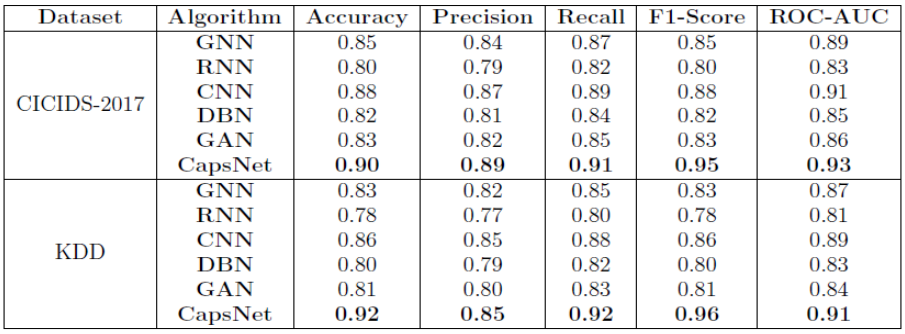


Algorithm 1 Deep Learning Model Optimization using Metaheuristic Algorithms

```
Input: Deep Learning Model (GNN, RNN, CNN, DBN, GAN, CapsNet)
Input: Metaheuristic Algorithms (SMA, GSK, RUN, HHO, WOA)
Input: Dataset
Input: Performance Metric (e.g., accuracy, precision, recall, F1-score, ROC-AUC)

Output: Optimized Model and Hyperparameters
Initialization:
Define hyperparameter ranges for each deep learning model.
Initialize metaheuristic algorithm parameters.
Set number of iterations and population size for each metaheuristic algorithm.
for each metaheuristic algorithm do
  for each deep learning model do
    Initialize Population:
    Initialize population with random hyperparameter sets.
    Evaluate Initial Population:
    Train the deep learning model with each hyperparameter set.
    Evaluate model performance using the selected metric.
    Optimization Loop:
    for each iteration do
      Generate New Solutions:
      Use metaheuristic algorithm to generate new hyperparameter sets.
      Update positions of individuals in the population.
      Evaluate New Solutions:
      Train the deep learning model with new hyperparameter sets.
      Evaluate model performance with the new hyperparameters.
      Update Best Solution:
      Track the best hyperparameter set and its performance.
      Update metaheuristic algorithm's best solution.
    end for
    Select the Best Model:
    Select hyperparameter set with best performance for the model.
  end for
end for
Final Model Selection:
Compare performance of models optimized by different metaheuristic algorithms.
Select model with best performance based on the chosen metric.
Output:
Return the deep learning model with the best hyperparameters found and its
performance metrics. =0
```

Table 1 Approximate Values of Evaluation Metrics for Deep Learning Algorithms on CICIDS-2017 and KDD Datasets

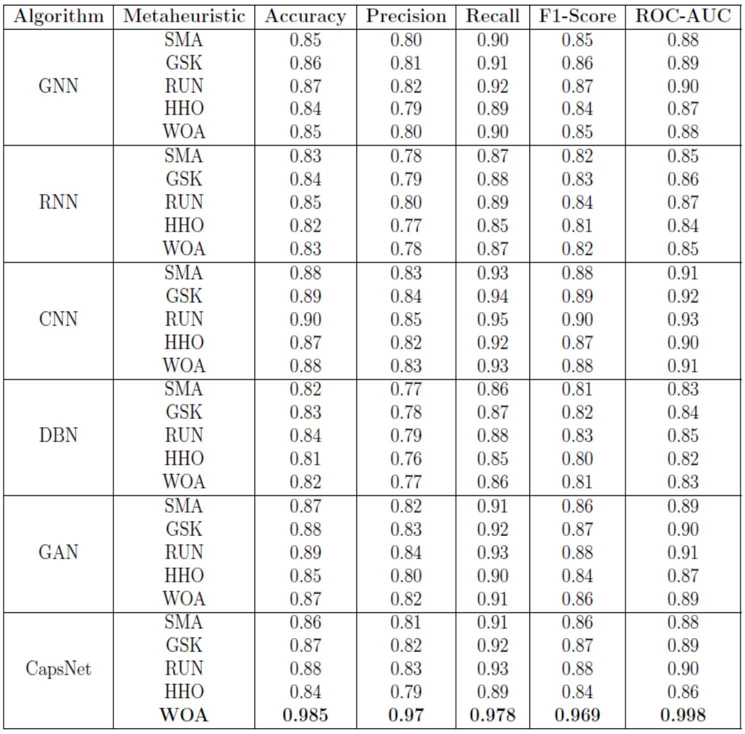


Dataset	Algorithm	Accuracy	Precision	Recall	F1-Score	ROC-AUC
CICIDS-2017	GNN	0.85	0.84	0.87	0.85	0.89
	RNN	0.80	0.79	0.82	0.80	0.83
	CNN	0.88	0.87	0.89	0.88	0.91
	DBN	0.82	0.81	0.84	0.82	0.85
	GAN	0.83	0.82	0.85	0.83	0.86
	CapsNet	**0.90**	**0.89**	**0.91**	**0.95**	**0.93**
KDD	GNN	0.83	0.82	0.85	0.83	0.87
	RNN	0.78	0.77	0.80	0.78	0.81
	CNN	0.86	0.85	0.88	0.86	0.89
	DBN	0.80	0.79	0.82	0.80	0.83
	GAN	0.81	0.80	0.83	0.81	0.84
	CapsNet	**0.92**	**0.85**	**0.92**	**0.96**	**0.91**

Journal of Information Systems Engineering and Management
2025, 10(34s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

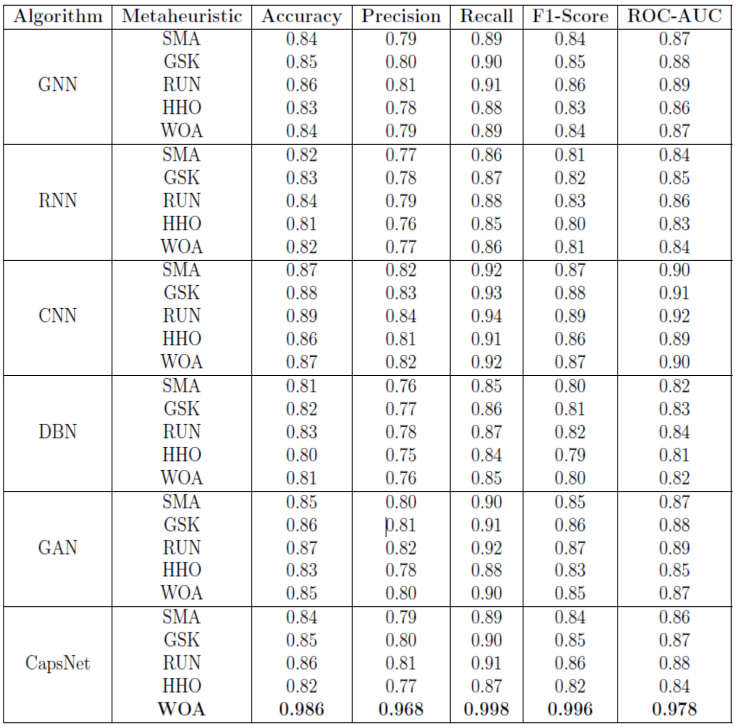
Research Article

Table 2 Approximate Performance Metrics for Deep Learning Algorithms Optimized with Metaheuristic Algorithms (CICIDS-2017 Dataset)



Algorithm	Metaheuristic	Accuracy	Precision	Recall	F1-Score	ROC-AUC
GNN	SMA	0.85	0.80	0.90	0.85	0.88
	GSK	0.86	0.81	0.91	0.86	0.89
	RUN	0.87	0.82	0.92	0.87	0.90
	HHO	0.84	0.79	0.89	0.84	0.87
	WOA	0.85	0.80	0.90	0.85	0.88
RNN	SMA	0.83	0.78	0.87	0.82	0.85
	GSK	0.84	0.79	0.88	0.83	0.86
	RUN	0.85	0.80	0.89	0.84	0.87
	HHO	0.82	0.77	0.85	0.81	0.84
	WOA	0.83	0.78	0.87	0.82	0.85
CNN	SMA	0.88	0.83	0.93	0.88	0.91
	GSK	0.89	0.84	0.94	0.89	0.92
	RUN	0.90	0.85	0.95	0.90	0.93
	HHO	0.87	0.82	0.92	0.87	0.90
	WOA	0.88	0.83	0.93	0.88	0.91
DBN	SMA	0.82	0.77	0.86	0.81	0.83
	GSK	0.83	0.78	0.87	0.82	0.84
	RUN	0.84	0.79	0.88	0.83	0.85
	HHO	0.81	0.76	0.85	0.80	0.82
	WOA	0.82	0.77	0.86	0.81	0.83
GAN	SMA	0.87	0.82	0.91	0.86	0.89
	GSK	0.88	0.83	0.92	0.87	0.90
	RUN	0.89	0.84	0.93	0.88	0.91
	HHO	0.85	0.80	0.90	0.84	0.87
	WOA	0.87	0.82	0.91	0.86	0.89
CapsNet	SMA	0.86	0.81	0.91	0.86	0.88
	GSK	0.87	0.82	0.92	0.87	0.89
	RUN	0.88	0.83	0.93	0.88	0.90
	HHO	0.84	0.79	0.89	0.84	0.86
	WOA	**0.985**	**0.97**	**0.978**	**0.969**	**0.998**

Table 3 Approximate Performance Metrics for Deep Learning Algorithms Optimized with Metaheuristic Algorithms (KDD Dataset)



Algorithm	Metaheuristic	Accuracy	Precision	Recall	F1-Score	ROC-AUC
GNN	SMA	0.84	0.79	0.89	0.84	0.87
	GSK	0.85	0.80	0.90	0.85	0.88
	RUN	0.86	0.81	0.91	0.86	0.89
	HHO	0.83	0.78	0.88	0.83	0.86
	WOA	0.84	0.79	0.89	0.84	0.87
RNN	SMA	0.82	0.77	0.86	0.81	0.84
	GSK	0.83	0.78	0.87	0.82	0.85
	RUN	0.84	0.79	0.88	0.83	0.86
	HHO	0.81	0.76	0.85	0.80	0.83
	WOA	0.82	0.77	0.86	0.81	0.84
CNN	SMA	0.87	0.82	0.92	0.87	0.90
	GSK	0.88	0.83	0.93	0.88	0.91
	RUN	0.89	0.84	0.94	0.89	0.92
	HHO	0.86	0.81	0.91	0.86	0.89
	WOA	0.87	0.82	0.92	0.87	0.90
DBN	SMA	0.81	0.76	0.85	0.80	0.82
	GSK	0.82	0.77	0.86	0.81	0.83
	RUN	0.83	0.78	0.87	0.82	0.84
	HHO	0.80	0.75	0.84	0.79	0.81
	WOA	0.81	0.76	0.85	0.80	0.82
GAN	SMA	0.85	0.80	0.90	0.85	0.87
	GSK	0.86	0.81	0.91	0.86	0.88
	RUN	0.87	0.82	0.92	0.87	0.89
	HHO	0.83	0.78	0.88	0.83	0.85
	WOA	0.85	0.80	0.90	0.85	0.87
CapsNet	SMA	0.84	0.79	0.89	0.84	0.86
	GSK	0.85	0.80	0.90	0.85	0.87
	RUN	0.86	0.81	0.91	0.86	0.88
	HHO	0.82	0.77	0.87	0.82	0.84
	WOA	**0.986**	**0.968**	**0.998**	**0.996**	**0.978**

Figure 4 Compare the Capsule Network (CapsNet) algorithm with and without optimization reveals significant differences in performance across two datasets: CICIDS-2017 and KDD. As a result of optimization with metaheuristic algorithms, CapsNet proves to be more effective than other algorithms. With optimization, CapsNet achieves an accuracy of 0.985, an F1-Score of 0.969, and a ROC-AUC of 0.998, compared to 0.90, 0.95, and 0.93 with non-optimized performance. As well, optimization yields an accuracy of 0.986, a F1-Score of 0.996, and a ROC-AUC of 0.978 on the KDD dataset, compared with an accuracy, F1-Score, and ROC-AUC of 0.92, 0.96, and 0.91, respectively, on the non-optimized dataset. The results indicate that CapsNet performs well in both conditions, however, its effectiveness is notably enhanced when it is optimized. Based on this observation, metaheuristic optimization has a significant role to play in refining the performance of CapsNet, demonstrating its robust capabilities as detailed in the preceding tables.

7 CONCLUSION

As a result of this study, we are able to demonstrate the benefits of integrating metaheuristic optimization techniques with deep learning techniques for enhancing intrusion detection systems. In combination with advanced models such as Capsule Networks, we have achieved notable improvements in detection accuracy and robustness of the system by combining Genetic Algorithms, Particle Swarm Optimization, and Harris Hawks Optimization. These results demonstrate the potential of metaheuristic-optimized deep learning approaches for reducing false positives and improving overall performance on benchmark datasets such as CICIDS-2017 and KDD. In this study, metaheuristics were used to fine-tune deep learning models in order to address the dynamic and complex nature of cyber threats. The field of intrusion detection can be advanced further by exploring additional optimization techniques and applying them to more diverse and challenging datasets in the future.

8 FUTURE WORK

It is possible to further enhance intrusion detection capabilities by exploring additional metaheuristic algorithms and their combinations in future research. Further insight may be gained by examining the impact of emerging deep learning architectures, such as Transformer-based models and Hybrid Neural Networks, on IDS performance. Moreover, the system would be more practical and robust if it was integrated with real-time data streams and evaluated in dynamic network environments. The proposed methods should also be evaluated in terms of scalability and comparison with other benchmark datasets. It may also be useful to examine the impact of adversarial attacks on the performance of metaheuristic-optimized IDS models in order to develop more resilient cybersecurity solutions. In order to bridge the gap between academic research and practical implementation, it would be beneficial to engage in collaborative research with industry partners to tailor these methods to specific applications and environments.

REFERENCES

[1] Bhumgara, A., Pitale, A.: Detection of network intrusions using hybrid intelligent system. In: International Conferences on Advances in Information Technology (2019)

[2] Kala, T.S., Christy, A.: An intrusion detection system using opposition based particle swarm optimization algorithm and pnn. In: International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (2019)

[3] Xu, C., Shen, J., Du, X., Zhang, F.: An intrusion detection system using a deep neural network with gated recurrent units. IEEE Access **6**, 48697–48707 (2018) https://doi.org/10.1109/ACCESS.2018.2864891

[4] Scarfone, K., Mell, P.: Guide to intrusion detection and prevention systems (idps). NIST Spec. Publ **800**, 94 (2007)

[5] Bhosale, K.S., Maria, A.P.: Data mining based advanced algorithm for intrusion detection in communication networks. In: International Conference on Computational Techniques, Electronics & Mechanical System (CTEMS) (2018)

[6] Belouch, M., El Hadaj, S., Idhammad, M.: Performance evaluation of intrusion detection based on machine learning using apache spark. Procedia Comput. Sci. **127**, 1–6 (2018) https://doi.org/10.1016/j.procs.2018.01.011

[7] Zhang, Z., Pan, P.: A hybrid intrusion detection method based on improved fuzzy c-means and svm. In: International Conference on Communication Information System and Computer Engineer [CISCE] (2019)

[8] Uikey, R., Cyanchandani, D.M.: Survey on classification techniques applied to intrusion detection system and its comparative analysis. In: 4th International Conference on Communication and Electronics System (ICCES 2019) (2019)

[9] Vinayakumar, R., Alazab, M., Soman, K., Poornachandran, P., Al-Nemrat, A., Venkatraman, S.: Deep learning approach for intelligent intrusion detection system. IEEE Access **7**, 41525–41550 (2019) https://doi.org/10.1109/ACCESS.2019. 2902400

[10] Cardoso, L.S.: Intrusion detection versus intrusion protection. In: Network Security: Current Status and Future Directions, pp. 99–115. IEEE Press(2007)

[11] Kemmerer, R.A., Vigna, G.: Intrusion detection: A brief history and overview. Computer **35**, 1012428 (2002) https://doi.org/10.1109/mc.2002.1012428

[12] Aminanto, E., Kim, K.: Deep learning in intrusion detection system: An overview. In: 2016 International Research Conference on Engineering and Technology (2016 IRCET) (2016). Higher Education Forum

[13] Shone, N., Ngoc, T.N., Phai, V.D., Shi, Q.: A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence **2**(1), 41–50 (2018)

[14] Belavagi, M.C., Muniyal, B.: Performance evaluation of supervised machine learning algorithms for intrusion detection. Procedia Comput. Sci. **89**, 117–123 (2016) https://doi.org/10.1016/j.procs.2016.06.020

[15] Javaid, A., Niyaz, Q., Sun, W., Alam, M.: A deep learning approach for network intrusion detection system. In: Proceedings of the 9th EAI International Conference on Bio-Inspired Information and Communications Technologies, pp. 21–26 (2016). https://doi.org/10.4108/eai.3-12-2015.2262789

[16] Almseidin, M., Alzubi, M., Kovacs, S., Alkasassbeh, M.: Evaluation of machine learning algorithms for intrusion detection system. In: 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), pp. 000277– 000282 (2017). IEEE

[17] Sohn, I.: Deep belief network based intrusion detection techniques: A survey. Expert Systems with Applications **167**, 114170 (2021)

[18] Saranya, T., Sridevi, S., Deisy, C., Chung, T.-D., Khan, M.A.: Performance analysis of machine learning algorithms in intrusion detection system: A review. Procedia Comput. Sci. **171**, 1251–1260 (2020) https://doi.org/10.1016/j.procs. 2020.04.219

[19] Liu, H., Lang, B.: Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences **9**(20), 4396 (2019)

[20] Susilo, B., Sari, R.F.: Intrusion detection in iot networks using deep learning algorithm. Information **11**(5), 279 (2020)

[21] Almomani, O.: A hybrid model using bio-inspired metaheuristic algorithms for network intrusion detection system. Computers, Materials & Continua **68**(1) (2021)

[22] Savanovi´c, N., Toskovic, A., Petrovic, A., Zivkovic, M., Damaˇseviˇcius, R., Jovanovic, L., Bacanin, N., Nikolic, B.: Intrusion detection in healthcare 4.0 internet of things systems via metaheuristics optimized machine learning. Sustainability **15**(16), 12563 (2023)

[23] Kareem, S.S., Mostafa, R.R., Hashim, F.A., El-Bakry, H.M.: An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection.
Sensors **22**(4), 1396 (2022)

[24] Alkanhel, R., El-kenawy, E.-S.M., Abdelhamid, A.A., Ibrahim, A., Alohali, M.A., Abotaleb, M., Khafaga, D.S.: Network intrusion detection based on feature selection and hybrid metaheuristic optimization. Computers, Materials & Continua **74**(2) (2023)

[25] Alhayali, R.A.I., Aljanabi, M., Ali, A.H., Mohammed, M.A., Sutikno, T.: Optimized machine learning algorithm for intrusion detection. Indonesian Journal of Electrical Engineering and Computer Science **24**(1), 590–599 (2021)

[26] Ghanbarzadeh, R., Hosseinalipour, A., Ghaffari, A.: A novel network intrusion detection method based on metaheuristic optimisation algorithms. Journal of Ambient Intelligence and Humanized Computing **14**(6), 7575–7592 (2023)

[27] Khafaga, D.S., Karim, F.K., Abdelhamid, A.A., El-kenawy, E.-S.M., Alkahtani, H.K., Khodadadi, N., Hadwan, M., Ibrahim, A.: Voting classifier and metaheuristic optimization for network intrusion detection. Computers, Materials & Continua **74**(2) (2023)

[28] Elmasry, W., Akbulut, A., Zaim, A.H.: Evolving deep learning architectures for network intrusion detection using a double pso metaheuristic. Computer Networks **168**, 107042 (2020)

[29] Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y.: A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems **32**(1), 4–24 (2020)

[30] Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M.: Graph neural networks: A review of methods and applications. AI open **1**, 57–81 (2020)

[31] Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: Lstm cells and network architectures. Neural computation **31**(7), 1235–1270 (2019)

[32] Staudemeyer, R.C., Morris, E.R.: Understanding lstm–a tutorial into long short-term memory recurrent neural networks. arXiv preprint arXiv:1909.09586 (2019)

[33] Sherstinsky, A.: Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. Physica D: Nonlinear Phenomena **404**, 132306
(2020)

[34] Kim, J., Kim, J., Kim, H., Shim, M., Choi, E.: Cnn-based network intrusion detection against denial-of-service attacks. Electronics **9**(6), 916 (2020)

[35] Vinayakumar, R., Soman, K., Poornachandran, P.: Applying convolutional neural network for network intrusion detection. In: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1222– 1228 (2017). IEEE

[36] Alom, M.Z., Bontupalli, V., Taha, T.M.: Intrusion detection using deep belief networks. In: 2015 National Aerospace and Electronics Conference (NAECON), pp. 339–344 (2015). IEEE

[37] Seo, E., Song, H.M., Kim, H.K.: Gids: Gan based intrusion detection system for in-vehicle network. In: 2018 16th Annual Conference on Privacy, Security and Trust (PST), pp. 1–6 (2018). IEEE

[38] Andresini, G., Appice, A., De Rose, L., Malerba, D.: Gan augmentation to deal with imbalance in imaging-based intrusion detection. Future Generation Computer Systems **123**, 108–127 (2021)

[39] Li, S., Chen, H., Wang, M., Heidari, A.A., Mirjalili, S.: Slime mould algorithm: A new method for stochastic optimization. Future generation computer systems **111**, 300–323 (2020)

[40] Gharehchopogh, F.S., Ucan, A., Ibrikci, T., Arasteh, B., Isik, G.: Slime mould algorithm: A comprehensive survey of its variants and applications. Archives of Computational Methods in Engineering **30**(4), 2683–2723 (2023)

[41] Mohamed, A.W., Hadi, A.A., Mohamed, A.K.: Gaining-sharing knowledge based algorithm for solving optimization problems: a novel nature-inspired algorithm. International Journal of Machine Learning and Cybernetics **11**(7), 1501–1529 (2020)

[42] Agrawal, P., Ganesh, T., Mohamed, A.W.: A novel binary gaining–sharing knowledge-based optimization algorithm for feature selection. Neural Computing and Applications **33**(11), 5989–6008 (2021)

[43] Ahmadianfar, I., Heidari, A.A., Gandomi, A.H., Chu, X., Chen, H.: Run beyond the metaphor: An efficient optimization algorithm based on runge kutta method. Expert Systems with Applications **181**, 115079 (2021)

[44] Shaban, H., Houssein, E.H., P´erez-Cisneros, M., Oliva, D., Hassan, A.Y., Ismaeel, A.A., AbdElminaam, D.S., Deb, S., Said, M.: Identification of parameters in photovoltaic models through a runge kutta optimizer. Mathematics **9**(18), 2313 (2021)

[45] Heidari, A.A., Mirjalili, S., Faris, H., Aljarah, I., Mafarja, M., Chen, H.: Harris hawks optimization: Algorithm and applications. Future generation computer systems **97**, 849–872 (2019)

[46] Alabool, H.M., Alarabiat, D., Abualigah, L., Heidari, A.A.: Harris hawks optimization: a comprehensive review of recent variants and applications. Neural computing and applications **33**, 8939–8980 (2021)

[47] Arnaout, J.-P.: A worm optimization algorithm to minimize the makespan on unrelated parallel machines with sequence-dependent setup times. Annals of Operations Research **285**(1), 273–293 (2020)

[48] Arnaout, J.-P.: Worm optimization: a novel optimization algorithm inspired by c. elegans. In: Proceedings of the 2014 International Conference on Industrial Engineering and Operations Management, Indonesia, pp. 2499–2505 (2014)