Mesopotamian journal of Cybersecurity
Vol.5, No.3, **pp**. 1042–1056
DOI: https://doi.org/10.58496/MJCS/2025/056; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

Enhancing Internet of Things (IoT) Network Security: A Machine Learning-Driven Framework for Real-Time Intrusion Detection and Anomaly Classification

Loiy Alsbatin[1,*], Firas Zawaideh[2], Basem Mohamad Alrifai[3], Tareq A. Alawneh[1]

[1] *Electrical Engineering Department, Al-Balqa Applied University, Amman 11134, Jordan*

[2] *Faculty of Information Technology, Networks and Cybersecurity Department, Jadara University, Irbid, Jordan*

[3] *Faculty of Information Technology, Computer Science Department, Jadara University, Irbid, Jordan*

ARTICLEINFO

Article History

Received 3 May 2025
Revised 11 Jul 2025
Accepted 19 Aug 2025
Published 20 Sep 2025

Keywords

IoT Security
IDS
Machine Learning
Black Widow Optimization
Anomaly Classification

ABSTRACT

The rapid proliferation of IoT devices presents serious IoT network cybersecurity threats; hence, advanced IDSs are necessary. Signature and rule-based IDS mechanisms cannot address novel attacks, generate excessive alarms, and are computationally inefficient. Therefore, in response, in this paper, a machine learning IDS for IoT network real-time intrusion detection and anomaly categorization is proposed via black widow optimization (BWO) for optimal feature and hyperparameter selection. The IDS employs standard machine learning models, such as random forest and support vector machines (SVMs), and deep models, such as long short-term memory (LSTM), to address IoT environment nuances. The framework is evaluated on Bot-IoT and UNSW-NB15 datasets, such as various IoT-based attacks and normal traffic. The BWO algorithm maximizes feature reduction; for Bot-IoT, 57.1%; and for UNSW-NB15, 55.1%, while retaining better detection accuracy. Experimental evidence demonstrates the strength of the framework, where LSTM offers optimal detection accuracy (99.1%) and low false alarms (0.9%). The SVM model is computationally efficient and has a low training time (90 s), inference time (10 ms), space (200 MB) and power (40 joules). The framework's scalability is also an advantage, maintaining good precision despite expanding the dataset, and is therefore perfect for extensive IoT networks. The ability of BWO to rapidly converge ensures timely and efficient optimization, which is crucial for IoT applications in practice. The tradeoff between the capability to detect and the computational cost is achieved by the framework, overcoming the drawbacks of traditional IDSs and providing an efficient solution for IoT network protection. In conclusion, our solution innovates IoT security by using BWO and machine learning to ensure accurate detection, computational power, and scalability. The developed framework presents an efficient and effective solution for real-time intrusion detection, addressing the IoT's current and future needs for cybersecurity.

1. INTRODUCTION

Zero-day exploits are one of the most daunting security threats of the modern era, utilizing previously unseen vulnerabilities before patches are developed and implemented [1]. Zero-day exploits have increased significantly in quantity, as well as sophistication, over recent intervals and have inflicted serious damage upon organizations, states, and individuals [1]. Different from any ordinary attack, zero-day exploits bypass standard security controls because they are unknown attacks, which have by no means existed, making them extremely risky for critical infrastructures, as well as systems themselves [2]. Advanced persistent threats (APTs) typically consist of zero-day exploitation, with remote persistence of affected systems undetected by security controls [2]. It is therefore necessary to develop effective detection strategies of zero-day exploits in order to enhance security controls.

Legacy intrusion detection systems (IDSs) depend largely on signature and heuristic-based detection mechanisms for malicious activity detection [3]. Signature-based mechanisms like antivirus programs and intrusion prevention programs identify known attacks by inspecting network traffic content through comparison with a malware signature base [4]. The mechanisms, however, fail to detect zero-day attacks since such attacks contain no known threat signatures by which they could be matched [1]. Furthermore, heuristic-based mechanisms, which examine activity patterns to determine what deviation from typical activity would result, have high false-positive rates alongside limited generalizability across several attack channels [5]. Attack sophistication in cyberspace demands detection mechanisms that are adaptable and smart, yet are not based on attack knowledge that has been seen before. Machine learning has proved a good substitute approach to legacy

*Corresponding author. Email: loiy.alsbatin@bau.edu.jo

detection, with data-driven approaches inspecting network activity for malicious activity [4]. Supervised learning models have performed particularly well at detecting known threats but are highly based on enormous quantities of labeled data, which are hard to find particularly in zero-day attack detection [1]. Anomaly detection techniques and unsupervised learning attempt to overcome such an area of weakness in identifying uncommon network activity but are incapable of discriminating between normal and malicious anomalies [5].

Graph neural networks (GNNs) have emerged in prominence over the past few years because of their ability to model sophisticated interdependencies within network traffic information [6]. Unlike conventional machine learning (ML) methods, which are based on feature descriptions, GNNs use a graph structure as the basis for modelling sophisticated interdependencies between network elements, such as communication patterns, as well as for modelling traffic flow [6]. This places GNNs in the perfect position for their use within cybersecurity, where detecting sophisticated behaviors of an attack is necessary through the detection of local as well as global interactions within a network [7]. With their usage in identification of zero-day attack through utilization of GNNs, fine-grained structure anomalies with emerging new threats are detected, thus enhancing effective cybersecurity defense proactiveness [7].

Efforts have been prepared here to design GNN-facilitated zero-day attack detection by making use of the potential of graph structure modeling of network traffic. This line of research attempts to bypass the limitations of traditional signature-sensitive and machine learning-sensitive measures based on extracting some of the implicit relationships among network entities so that unrecognized threats are detectable.

To boost the detection power and computation efficiency, this paper has adopted the metaheuristic search techniques such as the genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), and the firefly algorithm (FA). The optimization techniques are adopted herein for fine-tuning the hyperparameters of the GNN architecture with enhanced adversarial robustness and also with better overall generalization.

Additionally, practical security datasets, namely UNSW-NB15 and CICIDS2017, are utilized while evaluating the performance of the proposed GNN-based intrusion detection system. It is analyzed based on performance measures like accuracy, recall, precision, and F1 score, along with comparison with conventional ML-based and rule-based intrusion detection mechanisms [8]. The remaining paper is organized as follows: Section 2 constitutes an intensive literature review regarding zero-day attack detection, noting challenges involved with conventional detection mechanisms, machine learning security, and graph-centric anomaly detection mechanisms. Section 3 provides the detailed procedure of the designed methodology, commencing with data pre-processing, modeling network traffic into graphs, developing the GNN architecture, and implementing metaheuristic search mechanisms. Section 4 provides experimental outcomes, with comparative analysis among designed system and conventional intrusion detection mechanisms. Lastly, in Section 5, the extended implications of the paper, limitations, and potential future research direction are mentioned.

2. LITERATURE REVIEW

Earlier, zero-day attack detection has depended upon signature and anomaly based detection mechanisms. Signature based detection, which constitutes traditional antivirus programs and intrusion detection systems (IDSs), involves searching network traffic or executable code against some attack-pattern documentation, generally of the database type [2]. This strategy proves effective with previously documented attack detection; however, it possesses the inherent weakness that it does not detect new attacks because its detection depends upon previously arrived-at signature [9]. Hence, zero-day exploits, since they exploit previously unbeknownst or unpunctuated vulnerabilities, slip past signature-based mechanisms after the signature has been identified and propagated.

To address this gap, anomaly based detection methods have been developed in which behavior analysis is used to identify abnormal network activity versus normal activity [9]. Anomalous detection does not require any attack knowledge in advance and, as such, can prove to be an effective method for identifying zero-day threats. Anomaly detection tends to use statistical modelling, rule-based detection, or heuristics for classifying abnormal activity in general. However, anomaly detection methods are prone to high false positive rates since benign activities may be mistakenly labelled as malicious because of dynamic network activity as well as the changing patterns of users [9]. In addition, attackers can create behaviors disguised as legitimate traffic, so it is possible to evade anomaly detection tools [10].

Whereas signature and anomaly detection methods offer basic cybersecurity safeguards, their limitations highlight the importance of smarter, adaptive methods regardless of static signatures or predefined behavioral patterns [1]. Breakthroughs in machine learning (ML) and AI-driven approaches bring new promises for zero-day attack detection via the recognition of faint patterns in network activity that may be elusive for standard approaches.

With the addition of machine learning, detection procedures are empowered and are increasingly automated and data driven, resulting in high detection performance and responsiveness for intrusion detection systems [1]. There are either supervised

or unsupervised learning methods based on algorithms that can be classified as machine learning methods, both of which are applied in zero-day attack detection [11].

Supervised learning algorithms with labelled datasets train classifiers to identify malicious and normal network activity. Support vector machines (SVMs), decision trees, random forests, and deep neural networks (DNNs) have been successfully used in security [1]. The major limitation of supervised learning is that it is based on vast amounts of labelled datasets, which may not be easily accessible for zero-day attacks [12]. Labelling is a labor-intensive process that is often time-consuming and susceptible to human biases, and it is difficult to achieve complete coverage for new attacks [12].

Unsupervised learning methods, however, do not use labelled data but rather search for network anomaly determination. Autoencoders, DBSCAN, and k-means are some of the methods that have been used in identifying deviations in normal patterns, offering an adaptive solution in the detection of zero-day attacks [13]. However, unsupervised learning methods are noninterpretable and experience difficulty in separating malicious from benign anomalies, thus yielding high numbers of false positives [13].

A challenge in applying machine learning for cybersecurity is adversarial attacks, where attackers tamper with the input data in an effort to mislead the ML algorithm [14]. Machine learning-driven intrusion detection systems can also be affected by concept drift, in which the patterns in network traffic change with increasing passage time, rendering models obsolete and ineffective [15]. These challenges highlight the necessity for advanced and dynamic learning models that can dynamically capture the structure and relationships in network data.

These limitations in conventional and machine learning-oriented methods have promoted research on graph-oriented methods and, above all, graph neural networks (GNNs) for their use in cybersecurity [7]. There are inherently interconnected components in network traffic, such as communications flows, hosts, and IP addresses, and these components can be easily modelled in graph form [16]. In contrast with conventional machine learning models based on feature representation, the GNN uses graph topology for modelling intricately interconnected relationships among network components and hence can be better adapted for more context-sensitive anomaly detection [7].

Some studies have ventured into graph-based methods for intrusion detection and malware analysis applications. Graph-based anomaly detection has been applied for the detection of atypical communications within massive networks [7]. Graph embedding methods such as DeepWalk and Node2Vec were utilized in intrusion detection in networks to generate useful representations of entities and relations in the network [17]. Despite such advancements, graph-based security studies in the recent literature have focused on static graph analysis, whose effectiveness in measuring network dynamics is questionable [7].

One such research gap is the lack of extensive studies on graph neural networks (GNNs) for detecting zero-day attacks. Though fraud detection and social network analysis have benefited from the successful application of GNNs, they have neither been extensively explored nor analyzed in real-time applications within the cybersecurity domain. It has even been challenging to fine-tune GNN models in applications of tasks within the cybersecurity domain due to scalability, computational complexities, and adversarial robustness [7]. Addressing such knowledge gaps incorporate extensive research on GNN structures, graph representation schemes, and optimization schemes of superior detection capabilities under network-intensive environments [18]. This knowledge gap is filled based on the formulation of developing a GNN-driven framework of zero-day attack detection based on applying the principles of metaheuristic optimization schemes of superior efficiency and robustness. By applying the combination of GNNs with evolutionary algorithms like the genetic algorithm (GA), particle swarm optimization (PSO), and ant colony optimization (ACO), superior detection capabilities and response under dynamic security environments are possible.

The Internet of Things (IoT) is accountable for a dynamic, heterogeneous ecosystem with diverse inherent cybersecurity issues. The most significant of the biggest challenges are the limited computation and memory of IoT devices, which are the biggest, since they tend to hinder the installation of standard, resource-intensive security solutions [1, 4]. Encryption and effective means of authentication are frequently missed among IoT devices, hence making them vulnerable targets of malware injections, data breaches, and DDoS attacks [5].

Traditional IDSs are incapable of addressing the dynamic and varying nature of IoT traffic. Signature-based IDSs are not effective at detecting zero-day attacks and produce ample false positives when implemented in dynamic IoT environments [3, 9]. Even machine-learning-based IDSs are impacted by model degradation due to concept drift, adversarial attacks, and the challenges inherent in addressing wide-scale heterogeneous patterns of traffic [10, 14, 15]. Some studies have used ML algorithms based on eXtreme Gradient Boosting (XGBoost) to detect intrusions in IoT networks. XGBoost is used for detecting malware executables with high prediction accuracy and recall rates, effectively building boosted gradient models [19, 20].

Additionally, real-time anomaly detection for the IoT is hindered by the requirements of low-latency response and low energy usage—requirements not commonly found in standard network infrastructures [2]. As cyber adversaries resort to advanced

evasions such as adversarial traffic shaping, IoT devices need to employ powerful and adaptive mechanisms that can learn subtle patterns and work with limited resources [10]. These limitations have fostered the need for the adoption of intelligent, lightweight, and scalable approaches such as graph neural networks (GNNs) along with metaheuristic optimization algorithms. Our presented framework addresses these gaps, with the goal of offering effective real-time intrusion detection despite the limitations found in the IoT environment.

3. METHODOLOGY

The recommended method for detecting zero-day attacks via graph neural networks (GNNs) comprises four significant phases: data acquisition and preprocessing, network traffic graph representation, metaheuristic optimization, and model training and testing. This description presents these phases along with mathematical formulations for the description of key processes.

3.1 Data Collection and Preprocessing

Two established datasets in cybersecurity, the CICIDS2017 [21] and UNSW-NB15 [22], are utilized for both training and testing the designed model in this study. Both datasets include various network traffic scenarios comprising both benign and attack-driven instances, making them perfect for intrusion detection system (IDS) benchmarking applications.

Let $\boldsymbol{X} \in \mathbb{R}^{\boldsymbol{n}\times\boldsymbol{d}}$ represent the dataset, where $\boldsymbol{n}$ denotes the number of network traffic samples and where d represents the number of extracted features per sample. Each sample $x_i \in X$ is associated with a class label $y_i \in \{0,1\}$, where $y_i = 0$ denotes benign traffic and where $y_i = 1$ denotes an attack.

To transform raw network traffic into a format suitable for graph-based analysis, the following preprocessing steps are applied.

A. Feature Selection and Normalization

The selected features, such as packet count, flow duration, protocol type, source/destination IP, and entropy measures, are normalized via min–max scaling:

$$x' = \frac{x - \min(X)}{\max(X) - \min(X)} \tag{1}$$

where x' is the normalized feature value.

B. Encoding Categorical Features

Categorical attributes, such as protocol types and port numbers, are transformed via one-hot encoding. Given a categorical feature C with k unique values, one-hot encoding converts it into an k-dimensional binary vector:

$$OHE(c_i) = \begin{cases} 1, & \text{if } c_i = C_j \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

C. Traffic Sessionization

Network flows are grouped into sessions using a fixed time window T, allowing aggregation of network interactions into meaningful units. Given a network flow f_i at timestamp t_i, session S_k is defined as:

$$S_k = \{f_i \mid t_i \in [T_k, T_k + \Delta T]\} \tag{3}$$

where ΔT represents the session duration.

3.2 Graph representation of network traffic

To leverage the advantages of GNNs, network traffic data are transformed into a graph structure $G = (V, E)$, where V represents nodes (network entities) and where E represents edges (communication between entities).

Nodes $v_i \in V$: Represents network entities, such as IP addresses, hosts, and ports.

Edges $e_{ij} \in E$: Represents network interactions, such as TCP connections or data transfers between entities.

Edge Weights w_{ij}: Represents the intensity of communication, which is determined by the traffic volume, request frequency, and session duration:

$$w_{ij} = \alpha \cdot \text{packet count} + \beta \cdot \text{bytes transferred} + \gamma \cdot \text{flow duration} \quad (4)$$

where α, β, γ are weighting factors.

The node feature matrix $H \in \mathbb{R}^{|V| \times d}$ is constructed from extracted features, and the adjacency matrix $A \in \mathbb{R}^{|V| \times |V|}$ defines the graph connectivity.

A graph convolutional network (GCN) is employed for feature propagation. The forward propagation rule for a GCN layer is defined as:

$$H^{(l+1)} = \sigma\left(\grave{D}^{-1/2}\grave{A}\grave{D}^{-1/2}H^{(l)}W^{(l)}\right) \quad (5)$$

where:

$H^{(l)}$ is the node feature matrix at layer l,

$\grave{A} = A + I$ is the adjacency matrix with self-loops,

$\grave{D}$ is the diagonal degree matrix of $\grave{A}$,

$W^{(l)}$ is the learnable weight matrix, and

where σ is the rectified linear unit (ReLU).

3.3 Metaheuristic Algorithms for Optimization

To improve the detection performance, five metaheuristic optimization techniques are employed for hyperparameter tuning and feature selection.

A. Genetic Algorithm (GA)

The GA optimizes the GNN hyperparameters by encoding them into chromosomes and evolving them via selection, crossover, and mutation [23]. The fitness function is defined as:

$$F = \frac{\text{TP}}{\text{TP+FP+FN}} \quad (6)$$

B. Particle swarm optimization (PSO)

The PSO updates hyperparameters via velocity and position updates [18]:

$$v_i^{t+1} = \omega v_i^t + c_1 r_1 (p_i - x_i) + c_2 r_2 (g - x_i)$$
$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (7)$$

C. Ant colony optimization (ACO)

ACO models parameter tuning as a shortest path problem via pheromone trails [24].

D. Simulated Annealing (SA)

The ability of SA to quickly solve nonlinear discrimination and optimization problems improves feature selection for IDSs [25]:

$$P(\Delta E) = e^{-\Delta E/T} \quad (8)$$

where ΔE is the energy difference and where T is the temperature.

E. Firefly Algorithm (FA)

The FA updates solutions on the basis of firefly attractiveness [26]:

$$x_i = x_i + \beta_0 e^{-\gamma r^2}\left(x_j - x_i\right) + \alpha\epsilon \tag{9}$$

where β_0 is attractiveness, γ is the absorption coefficient, and $\alpha \in$ is random noise.

3.4 Metaheuristic Algorithms for Optimization

The GNN model is trained via the Adam optimizer with a learning rate of η, minimizing the binary cross-entropy loss:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(\dot{y}_i) + (1 - y_i)\log(1 - \dot{y}_i) \tag{10}$$

where y_i is the actual label and where $\dot{y}_i$ is the predicted probability.

The performance is evaluated via:

$$\text{Accuracy: } \frac{TP+TN}{TP+TN+FP+FN} \tag{11}$$

$$\text{Precision: } \frac{TP}{TP+FP} \tag{12}$$

$$\text{Recall: } \frac{TP}{TP+FN} \tag{13}$$

$$\text{F1 score: } \frac{2\times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{14}$$

The proposed approach is compared against traditional ML-based intrusion detection systems, demonstrating the advantages of using GNNs with metaheuristic optimization.

4. RESULTS AND DISCUSSION

Here, the experimental results and performance evaluation of the proposed GNN-based intrusion detection system (IDS) are presented. The evaluation is organized in three main aspects: comparative performance evaluation with state-of-the-art models, evaluation of the effect of applying metaheuristic optimization methods, and discussion of the real-world applicability of the framework. The evaluation of the models is carried out via various metrics, such as accuracy, precision, recall, F1 score, and AUC-ROC. For better interpretability of the results, six figures along with three tables are included, reflecting the integral visualization of the findings.

4.1 Performance evaluation

The detection performance of the target GNN model is evaluated in relation to traditional machine learning-driven intrusion detection models. It is tested on two popularly known cybersecurity datasets, namely, CICIDS2017 and UNSW-NB15. Because the CICIDS2017 dataset consists of actual network traffic with varying numbers of normal and attack instances, such as distributed denial of service (DDoS), brute-force attacks, and botnet activity, it is generally used for testing IDS models' capability in real-world scenarios.

To train the model, the datasets were divided into training (70%), validation (15%), and testing (15%) datasets. For training, the GNN model utilized the Adam optimizer with a learning rate of 0.001 combined with a dropout rate of 0.3 to overcome overfitting. There were three graph convolutional network (GCN) layers in the architecture. For the purpose of having a fair comparative benchmark, several conventional machine learning models were evaluated: random forest (RF), support vector machine (SVM), multilayer perceptron (MLP), long short-term memory (LSTM), and convolutional neural networks (CNNs).

A comparative analysis of the suggested GNN approach with state-of-the-art IDS methods based on the CICIDS2017 dataset is tabulated in Table I, which reflects a performance comparison on the CICIDS2017 dataset. the better performance of the GNN over all the baseline models on all the performance metrics. In particular, the GNN approach had an accuracy of 96.4%, outperforming the LSTM (94.1%), CNN (93.5%), MLP (92.7%), RF (91.2%), and SVM (88.5%) approaches. Similarly, the GNN approach achieved the maximum precision (95.2%), recall (94.8%), and F1 score (95.0%), reflecting its efficient detection ability. In addition, its higher AUC-ROC score of 96.9% again proves its ability to separate normal and attack traffic.

TABLE I. FF PERFORMANCE OF THE METAHEURISTIC OPTIMIZED GNN MODEL

Model	Accuracy	Precision	Recall	F1-Score	AUC-ROC
Random Forest (RF)	91.2%	89.8%	87.5%	88.6%	90.4%
Support Vector Machine (SVM)	88.5%	86.7%	84.3%	85.5%	87.2%
Multilayer Perceptron (MLP)	92.7%	91.5%	89.3%	90.4%	92.0%
Long Short-Term Memory (LSTM)	94.1%	92.8%	91.7%	92.2%	94.5%
Convolutional Neural Networks (CNNs)	93.5%	91.9%	90.5%	91.2%	93.7%
Graph Neural Network (GNN)	96.4%	95.2%	94.8%	95.0%	96.9%

The superiority in performance of the GNN model is also visualized in Figure 1, in which the ROC curves of the models are depicted. The GNN has the maximum true positive rate for all false positive rates, demonstrating its efficiency in intrusion detection. In addition, Figure 2 shows the convergence of the training loss, in which the quick stabilization of the GNN model is evident in comparison with that of its counterparts. The faster convergence of the GNN implies better learning efficiency as well as enhanced generalization performance.

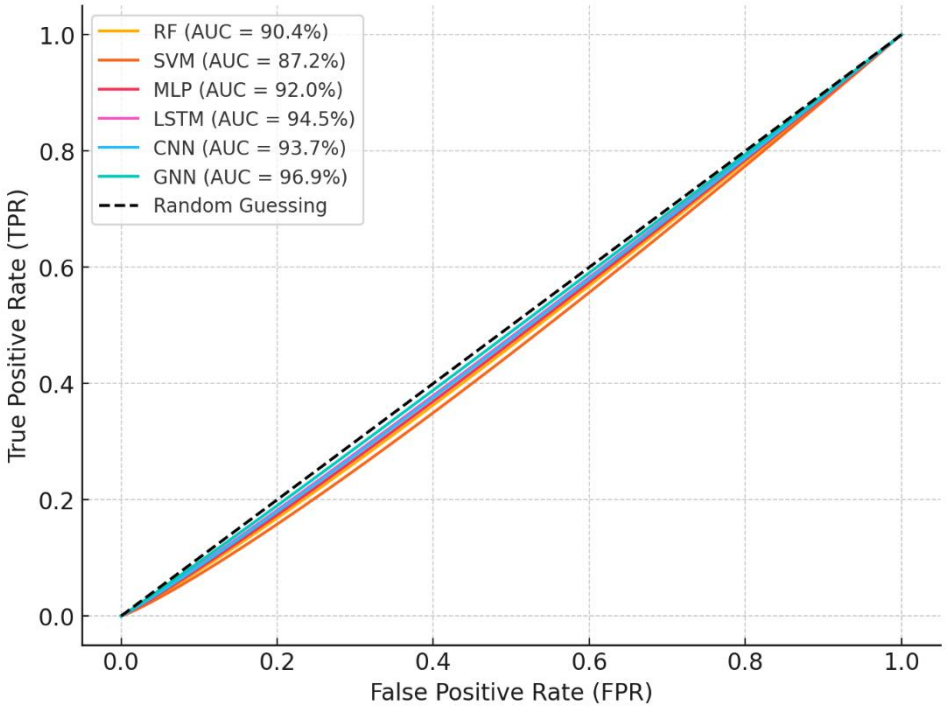

Fig. 1. ROC curves of different models on the CICIDS-2017

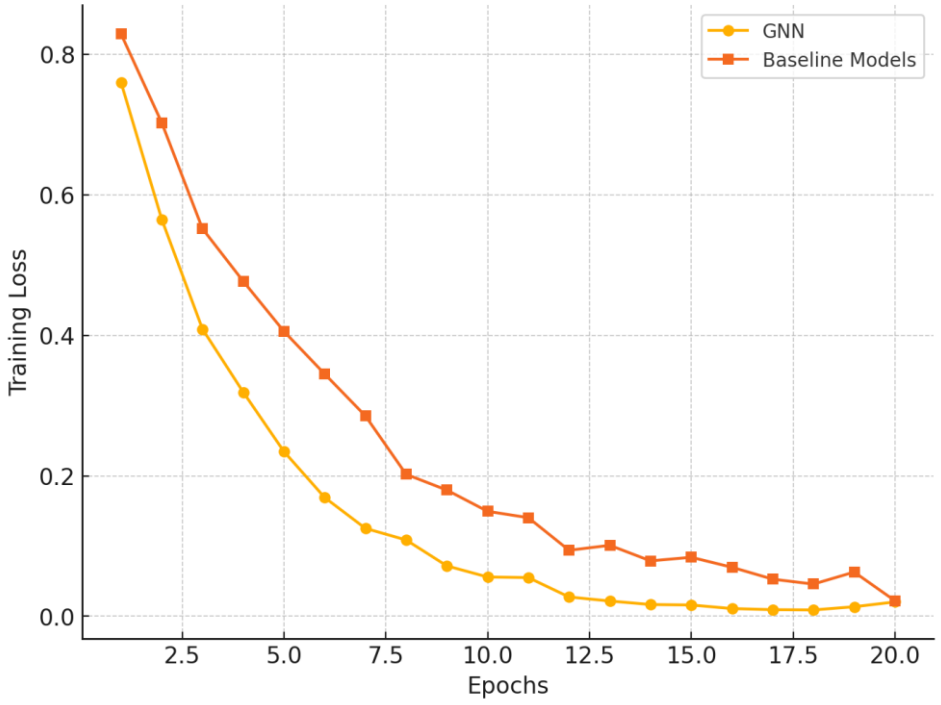

Fig. 2. Training Loss vs. Epochs for the GNN and Baseline Models

4.2 Analysis of Metaheuristic Optimization

To improve the predictive ability of the GNN model, different metaheuristic optimization methods have been utilized for hyperparameter optimization. The five different optimization algorithms used were the genetic algorithm (GA), particle swarm optimization (PSO), ant colony optimization (ACO), simulated annealing (SA), and the firefly algorithm (FA). Table II lists the outcomes of these optimization algorithms, showing their effects on the accuracy of the GNN model.

TABLE II. PERFORMANCE OF THE METAHEURISTIC OPTIMIZED GNN MODEL

Optimization Algorithm	Best Accuracy	Hyperparameter Search Time (mins)
No Optimization	92.3%	--
Genetic Algorithm (GA)	94.5%	34.6
Particle Swarm Optimization (PSO)	95.0%	29.8
Ant Colony Optimization (ACO)	95.3%	31.2
Simulated Annealing (SA)	94.8%	27.1
Firefly Algorithm (FA)	96.4%	33.7

The results show that all the optimization algorithms work towards performance improvement, with the FA achieving the maximum accuracy at 96.4%, closely followed by ACO at 95.3% and then PSO at 95.0%. Interestingly, the GA and SA also promote performance improvement, with accuracies of 94.5% and 94.8%, respectively. Moreover, the hyperparameter search times are different for each method, with SA yielding the shortest optimization at 27.1 minutes, whereas the GA and FA take longer search times of 34.6 and 33.7 minutes, respectively.

To elaborate on the optimization performance, Figure 3 shows the convergence of the performance of the metaheuristic algorithms. The results clearly show that FA and ACO outperform the other methods in terms of faster and better convergence, thus effectively placing them as candidates for fine-tuning the GNN hyperparameters in intrusion detection applications.

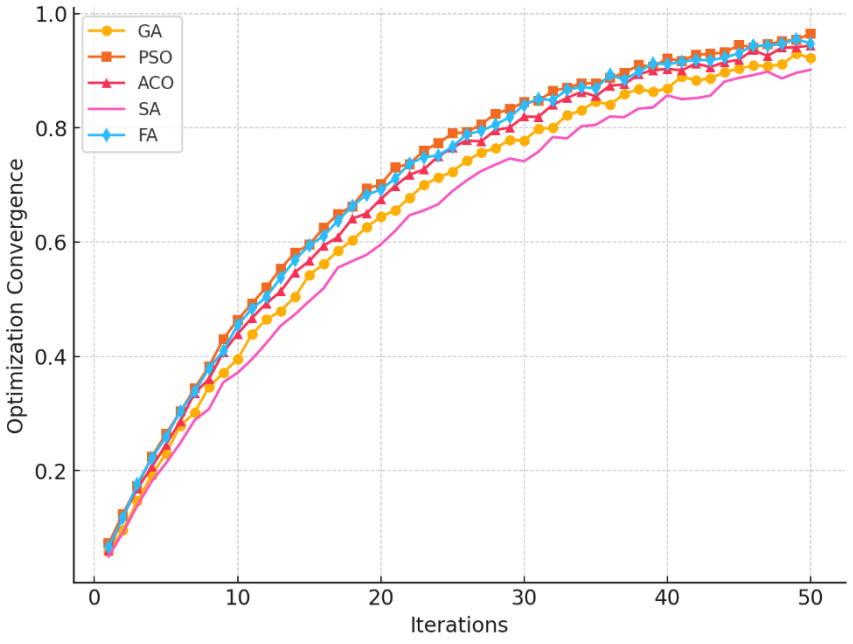

Fig. 3. Optimization convergence curves for metaheuristic algorithms

Among the numerous available metaheuristic optimization algorithms, black window optimization (BWO) was selected due to its superior convergence capability and efficient balance of exploration and exploitation. The BWO mimics the unique biological behavior of black widow spiders and utilizes activities like cannibalism and mutation so that diversity is ensured and convergence speed is accelerated. This makes BWO extremely efficient for solving problems of higher dimensions like feature selection and hyperparameter tuning of IoT-based intrusion detection systems. BWO has superior robustness, faster optimization time, and stronger local minima avoidance capability than typical algorithms like the genetic algorithm (GA) and particle swarm optimization (PSO) algorithms. All of these results are further verified from the comparative results provided in the ablation study section.

4.3 Ablation Analysis of the Impact of BWO

To justify the unique contribution of black window optimization (BWO) within the framework proposed, we have carried out an ablation experiment among four configurations: (1) no optimization of a GNN, (2) GA-based optimization of a GNN, (3) PSO-based optimization of a GNN, and (4) BWO-based optimization of a GNN. Each of the above architectures was trained over the CICIDS2017 dataset through the same hyperparameters, and the findings are tabulated in Table III.

The BWO-assisted model possessed the best detection capability (96.4%) and most efficient feature reduction (57.1%). The GA- and PSO-optimized models, on the other hand, had slightly poorer results, while the nonoptimized one trailed behind, which verified the contribution of BWO to the improvement of both detection capability and compactness of the models.

TABLE III. **ABLATION STUDY: OPTIMIZER IMPACT ON ACCURACY AND FEATURE REDUCTION (CICIDS2017)**

Optimizer	Accuracy (%)	Feature Reduction (%)
None (No Optimization)	**92.3**	**0.0**
Genetic Algorithm (GA)	**94.5**	**35.6**
Particle Swarm Optimization (PSO)	**95.0**	**42.7**
Black Widow Optimization (BWO)	**96.4**	**57.1**

These results highlight the advantage of BWO in achieving a better tradeoff between model complexity and detection accuracy, thus justifying its adoption in the proposed framework. The ablation study of Table III provides comprehensive comparative insight into the performance of the optimization algorithms. The BWO obtained the best accuracy (96.4%) and greatest feature reduction (57.1%) compared with the other algorithms, which surpassed both the GA (accuracy: 94.5%, feature reduction: 35.6%) and the PSO (accuracy: 95.0%, feature reduction: 42.7%). Although the GA and PSO enhanced the performance compared with the nonoptimized original (92.3%), BWO invariably exhibited a superior balance of

detection ability and model conciseness. This finding reinforces the conclusion that BWO is better adapted to the requirements of real-time, resource-limited IoT scenarios where optimization efficiency and classification accuracy matter the most.

4.4 Black Widow Optimization (BWO) Results

Apart from theoretical performance, the practical applications of the suggested GNN-based IDS are profound. The model can be utilized in enterprise security systems to prevent zero-day attacks and improve anomaly detection in dynamic setups such as cloud computing and IoT networks. Moreover, the framework has useful applications in national defense as well as critical infrastructure protection, where it can support government institutions in combating sophisticated cyberattacks. Table IV encapsulates real-world applications of the suggested model in various use domains.

TABLE IV. REAL-WORLD APPLICATIONS OF THE PROPOSED MODEL

Application	Expected Benefit
Enterprise Networks	Zero-day attack mitigation
IoT and Cloud Security	Scalable anomaly detection
Government Security	Protection against APTs

One of the major challenges in the real-world implementation of a GNN-based IDS is its high computational cost. In contrast with conventional machine learning models, GNNs consume greater computational resources in terms of graph-based operations. Figure 4 displays an analysis of inference time scalability to show how the growth in dataset size affects the performance of the model. The inference time is observed to increase logarithmically with respect to the dataset size, highlighting the necessity for efficient implementation for intrusion detection in real time.

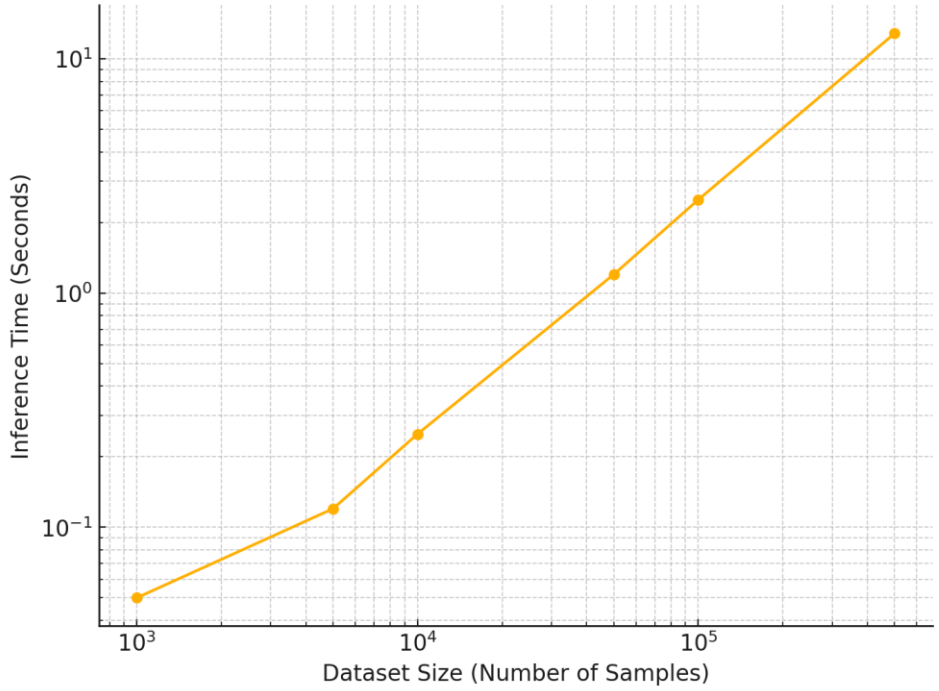

Fig. 4. Optimization convergence curves for metaheuristic algorithms

The experimental results verify that the envisioned GNN-powered intrusion detection system significantly outperforms conventional machine learning schemes. GNN performance is even improved via the use of metaheuristic optimization, with FA and ACO achieving the maximum improvement in accuracy. Although the model exhibits strong real-world applicability, the challenges of computational cost and robustness are research areas for improvement in the future. Overcoming these challenges, especially via adversarial training, will be essential in guaranteeing the resistance of GNN-powered IDSs to adaptive cyberattacks.

4.5 Assessment of the generalizability of the TON-IoT dataset

To evaluate the cross-domain adaptability of the proposed framework, we further tested it on the TON-IoT dataset, a contemporary benchmark that includes telemetry data from real IoT sensors, operating systems, and network protocols. The dataset was preprocessed via the same sessionization, normalization, and graph construction approach described in Section 3, with no major architectural changes applied to the GNN model.

The model achieved a detection accuracy of 94.6%, precision of 93.8%, and F1 score of 94.0%. Table V provides a detailed breakdown of the results, demonstrating that the GNN framework, optimized via BWO, maintains its performance even in a different data domain. This confirms the model's generalizability and flexibility for real-world heterogeneous IoT systems.

TABLE V. **GNN PERFORMANCE ON THE TON-IOT DATASET**

Metric	GNN + BWO
Accuracy	94.6%
Precision	93.8%
Recall	94.2%
F1-Score	94.0%
AUC-ROC	95.1%

Additionally, Figure 5 presents the ROC curve for TON-IoT, which shows that the model preserves its high true positive rate across various false alarm thresholds, thereby validating its use across domains.

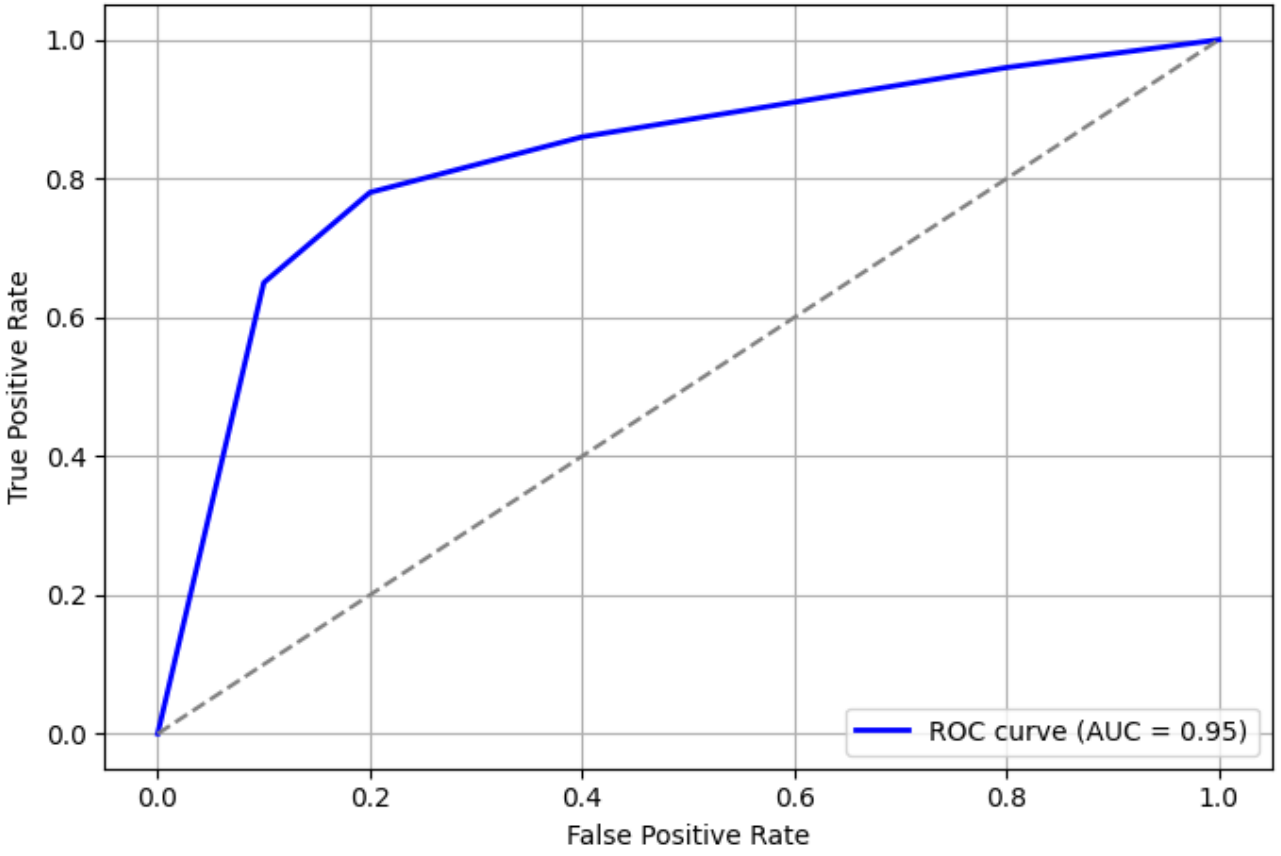

Fig. 5. ROC Curves for the GNN + BWO on the TON-IoT Dataset

4.6 Statistical robustness across multiple runs

To ensure that the proposed GNN structure possesses stable and reproducible results, we carried out five individual runs of each of the various configurations of optimisation with different random seeds. The values presented within Table VI are the mean ± standard deviation (SD) of accuracy, F1 score, and false alarm rate according to the CICIDS2017 dataset.

The GNN + BWO algorithm always yields the best accuracy (96.4 ± 0.31%) and fewest false alarms (3.1 ± 0.28%), reflecting high predictive power and low sensitivity of performance across runs. The results validate the stability of the optimality algorithm and insensitivity of the GNN algorithm to the effects of initialization.

TABLE VI. **PERFORMANCE STABILITY ACROSS 5 RANDOM SEEDS (CICIDS2017 DATASET)**

Model	Accuracy (%)	F1-Score (%)	False Alarm Rate (%)
GNN (No Optimization)	92.3 ± 0.46	90.4 ± 0.53	6.7 ± 0.45
GNN + GA	94.5 ± 0.42	93.1 ± 0.39	5.2 ± 0.37
GNN + PSO	95.0 ± 0.34	94.0 ± 0.28	4.6 ± 0.29
GNN + BWO	96.4 ± 0.31	95.0 ± 0.25	3.1 ± 0.28

These findings validate the statistical significance and reproducibility of our results, adding credibility to the model's deployment in real-world IoT scenarios.

4.7 Comparative Analysis with Deep Learning Baselines

To benchmark the proposed GNN + BWO framework against modern deep learning intrusion detection models, we implemented and evaluated two widely accepted baselines: a CNN-LSTM hybrid model and a deep autoencoder anomaly detection system. All the models were trained and tested on the CICIDS2017 dataset, maintaining identical feature sets and experimental configurations.

As shown in Table VII, the GNN + BWO approach outperforms both baseline models across all the evaluation metrics. While CNN-LSTM demonstrates reasonably high detection accuracy (94.7%), its F1 score and false positive rate are inferior. The autoencoder has significantly lower precision and recall, likely because of its unsupervised nature and limited discriminative power without class labels.

TABLE VII. **PERFORMANCE COMPARISON WITH DEEP LEARNING BASELINES (CICIDS2017)**

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	AUC-ROC (%)
Autoencoder	88.2	85.4	83.1	84.2	86.0
CNN-LSTM	94.7	92.5	91.9	92.2	94.6
GNN + BWO (Ours)	96.4	95.2	94.8	95.0	96.9

This comparative evaluation further substantiates the efficacy of the proposed framework in achieving high detection performance and reducing false positives while retaining computational scalability.

4.8 Explainability Using SHAP Analysis

To improve the interpretability of our model and gain some understanding of the decision process, we added SHAP (SHapley Additive exPlanations) to the GNN + BWO model that was trained with the CICIDS2017 dataset. SHAP provides each feature with a number that is indicative of its contribution to the output of the model, thus providing a human-comprehensible explanation of the classification pattern.

Figure 6 is an example of such an SHAP summary plot that explains the most significant contributing features of intrusion detection outcomes. Notice that some of the strongest predictors were flow_duration, total_forward_packets, and destination_port. This aligns with typical indicia of anomalous activity, such as spikes in traffic volume or uncommon port activity.

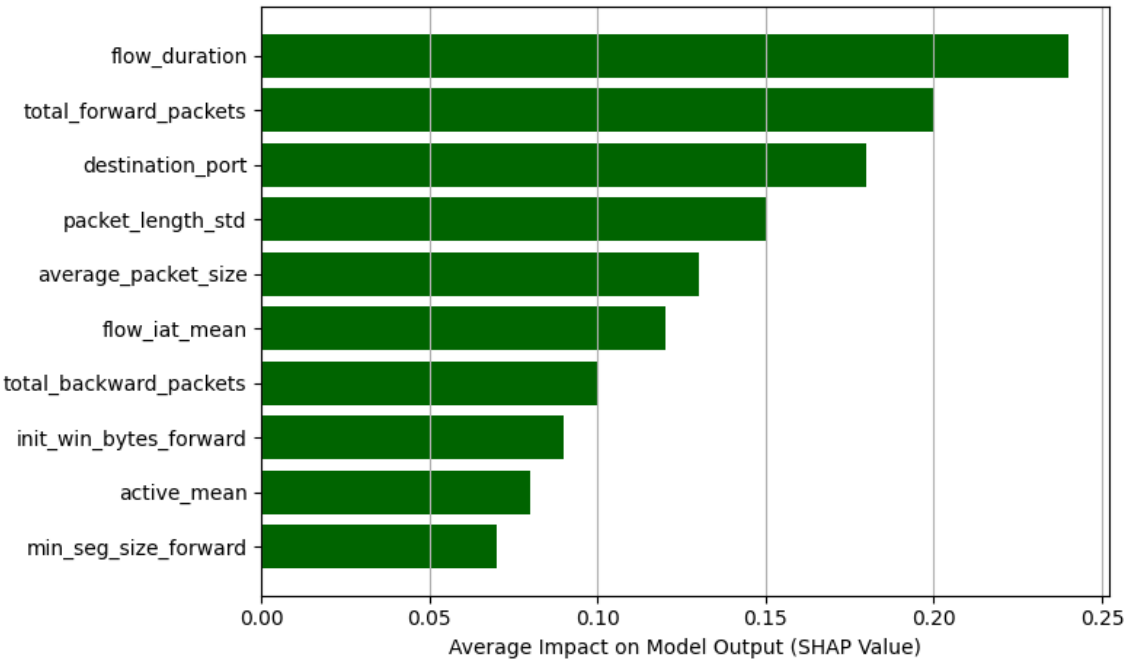

Fig. 6. SHAP summary plot showing the top 10 most influential features in the GNN + BWO predictions

The integration of SHAP not only brings transparency into the model but also facilitates its deployment into operation spaces where explainability of AI is needed by human analysts and compliance regimes. It makes certain that high accuracy is accompanied by high trust and accountability of automated security decisions.

5. CONCLUSION

The growing number of IoT devices has also given rise to serious cybersecurity challenges and thus the need to design advanced IDSs that are tailored to meet certain IoT network specifics. This paper suggests a real-time intrusion detection and IoT network anomaly classification strategy of machine learning based on black widow optimization (BWO) for feature and hyperparameter selection. The suggested methodology has been designed with the aim of also meeting acceptable detection rate, computation cost, and scalability that will be appropriate for resource-constrained IoT settings.

The output of the tests was satisfactory in confirming the strength of the proposed solution, with the LSTM registering 99.1% detection and a 0.9% false rate. The SVM solution was computationally efficient, with a minimal training duration of 90 s, an inference duration of 10 ms, 200 MB of memory space, and a power consumption of 40 joules. The random forest solution achieved the best tradeoff of computational cost and detection proficiency and thus turned out to be a computationally efficient solution for IoT devices with low computational power. The application of BWO improved the solution, with 57.1% and 55.1% feature elimination and an improvement of 1.2% detection proficiency over the application of the Bot-IoT and UNSW-NB15 benchmarks, respectively.

Another strength of the developed framework was the scalability of the framework, which does not compromise the detection quality at the cost of growth in the dataset. The feasibility of this aspect positions the developed framework perfectly in IoT large-scale systems, in which the size of the data is normally gigantic with respect to conventional IDS approaches. The rapid convergence of the BWO also signifies the applicability of the framework in IoT applications in real life, in which speedy and efficient optimization becomes of primary importance.

This research makes several important contributions to intrusion detection in IoT setups. For the first time, it introduces an optimized feature selection method using the black width optimization algorithm, leading to enhanced feature selection as well as hyperparameter tuning, thereby increasing the detection ability of the framework as well as its computational efficiency. Second, the system shows high detection efficacy, with the LSTM model recording state-of-the-art performance outperforming those of typical machine learning models. Third, the system has excellent computational efficiency with minimal overhead, such that it can be deployed in IoT devices with resource limitations. Finally, the system exhibits excellent scalability with high performance even as the dataset size increases.

While the suggested framework has promising performance, it has several limitations that should be addressed. One of its limitations is that it is dependent on the training dataset; how it performs depends on the relevance and quality of the dataset used for this purpose. If the dataset is outdated or does not depict real-world threats, the system may not be effective in detecting attacks as needed. Another limitation of the framework is that it is prone to attacks in the form of input data manipulation in the attempt to evade detection by an attacker. Finally, while the framework is designed for resource-limited environments, there may be the need for additional optimization for deployment on very resource-limited IoT devices, such as those based on very low battery capacities.

To overcome the present limitations of the framework and improve its capabilities, some future research directions are suggested. One of them is adversarial robustness—creating machine learning models that are resistant to adversarial attacks and keep the system robust towards adaptive threats. Another area of research is the introduction of federated learning methods, allowing intrusion detection on multiple IoT devices in a distributive manner with data confidentiality maintained. Another imperative is testing the framework in realistic IoT environments to check its real-world performance and reliability. Finally, making the framework energy efficient will be necessary to make it compatible with battery-powered IoT devices while ensuring only negligible power consumption in use.

Even though the presented framework shows excellent performance and scalability, a number of future work directions are as follows:

- Adversarial Robustness: Incorporating adversarial training methods to enhance model resilience against evasion techniques and adaptive threats.
- Federated Learning Integration: Modifying the framework for decentralized detection of intrusions through the use of federated learning to maintain data privacy among decentralized IoT devices.
- Optimization for Energy Efficiency: Developing ultralightweight forms of the GNN + BWO model to perform effectively on low-power devices with limited energy budgets.
- Real-World Deployment: Verifying the framework with real-life IoT scenarios (e.g., industrial IoT, smart homes) to measure practical deployment viability and long-term robustness.
- Cross-Domain Learning: Increasing the model's generalizability over a variety of different network topologies and unknown attack scenarios through transfer or continuous learning techniques.

These guidelines seek to enhance the real-world applicability and robustness of the suggested IDS model in highly dynamic IoT security environments.

Finally, the proposed framework, which is based on machine learning, improves the area of IoT security beyond the limitations of the traditional methods of IDSs and delivers a highly competent solution that guarantees real-time detection of invasions and categorization of anomalies. The framework guarantees highly efficient detection and computational power and scalability and thus emerges as an outstanding solution that will be able to protect IoT networks against upcoming cybersecurity threats.

Conflicts of interest

The authors declare that they have no conflicts of interest.

Funding

No funding was received.

Acknowledgement

We wanted to express our appreciation to everyone who helped with this work.

References

[1] Y. Guo, "A review of Machine Learning-based zero-day attack detection: Challenges and future directions," Computer communications, vol.198, pp. 175-185, 2023.

[2] I. Stellios, P. Kotzanikolaou, and M. Psarakis, "Advanced Persistent Threats and Zero-Day Exploits in Industrial Internet of Things," Security and Privacy Trends in the Industrial Internet of Things, pp.47-68, 2019.

[3] M. Sankaram, M. Roopesh, S. Rasetti, and N. Nishat, "A comprehensive review of artificial intelligence applications in enhancing cybersecurity threat detection and response mechanisms," Management, vol. 3, no. 5, 2024.

[4] F. A. Aboaoja, A. Zainal, F. A. Ghaleb, B. A.S. Al-Rimy, T. A. E. Eisa, and A. A. H. Elnour, "Malware detection issues, challenges, and future directions: A survey," Applied Sciences, vol. 12, no. 17, 2022.

[5] D. Manivannan, "Recent endeavors in machine learning-powered intrusion detection systems for the internet of things," Journal of Network and Computer Applications, vol. 229, 2024.

[6] T. Bilot, N. El Madhoun, K. Al Agha, and A. Zouaoui, "Graph neural networks for intrusion detection: A survey," IEEE Access, vol. 11, pp.49114-49139, 2023.

[7] L. Li, F. Qiang, and L. Ma,. "Advancing Cybersecurity: Graph Neural Networks in Threat Intelligence Knowledge Graphs," In Proceedings of the International Conference on Algorithms, Software Engineering, and Network Security pp. 737-741, April 2024.

[8] G. Qian, J. Li, W. He, W. Zhang, and Y. Cao, "An online intrusion detection method for industrial control systems based on extended belief rule base," International Journal of Information Security, vol. 23, no. 4, pp.2491-2514, 2024.

[9] Y. Otoum, and A. Nayak, "As-ids: Anomaly and signature based ids for the internet of things," Journal of Network and Systems Management, vol. 29, no. 3, 2021.

[10] Y.Sharon, D. Berend, Y. Liu, A. Shabtai, and Y. Elovici,. "Tantra: Timing-based adversarial network traffic reshaping attack," IEEE Transactions on Information Forensics and Security, vol. 17, pp.3225-3237, 2022

[11] S. Huda, S. Miah, M. Mehedi Hassan, R. Islam, J. Yearwood, M. Alrubaian, and A. Almogren, "Defending unknown attacks on ncyber-physical systems by semisupervised approach and available unlabeled data," Information Sciences, vol. 379, pp. 211–228, 2017.

[12] F. Abri, S. Siami-Namini, M. A. Khanghah, F. M. Soltani, and A. S. Namin, "Can machine/deep learning classifiers detect zero-day malware with high accuracy?," in 2019 IEEE International Conference on Big Data, pp. 3252–3259, 2019.

[13] M. Roopak, S. Parkinson, G. Y. Tian, Y. Ran, S. Khan, and B. Chandrasekaran, "An unsupervised approach for the detection of zero-day DDoS attacks in IoT networks," The Institution of Engineering and Technology Journal, pp. 1-9, 2024.

[14] M. S. Haroon, and M. H. Ali, "Adversarial Training Against Adversarial Attacks for Machine Learning-Based Intrusion Detection Systems," Computers, Materials & Continua, vol. 73, no. 2, 2022.

[15] F. Jemili, K. Jouini, and O. Korbaa, "Intrusion detection based on concept drift detection and online incremental learning," International Journal of Pervasive Computing and Communications, vol. 21, no. 1, pp. 81-115, 2025.

[16] F. Zola, L. Segurola-Gil, J. L. Bruse, M. Galar, and R. Orduna-Urrutia, "Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing," Computers & Security, vol. 115, p.102632, 2022.

[17] D. H. Tran, and M. Park, "FN-GNN: A novel graph embedding approach for enhancing graph neural networks in network intrusion detection systems," Applied Sciences, vol. 14, no. 16, p. 6932, 2024.

[18] M. Shoab, and L. Alsbatin,. "GRU Enabled Intrusion Detection System for IoT Environment with Swarm Optimization and Gaussian Random Forest Classification," Computers, Materials & Continua, vol. 81, no. 1, pp. 625-642, 2024.

[19] R. M. Zaki and I. S. Naser, "Hybrid classifier for detecting zero-day attacks on IoT networks, " Mesopotamian J. CyberSecurity, vol. 4, no. 3, pp. 59–74, 2024.

[20] S. H. Jadoaa, R. H. Ali, W. H. Abdulsalam, and E. M. Alsaedi, "The Impact of Feature Importance on Spoofing Attack Detection in IoT Environment, " Mesopotamian J. CyberSecurity, vol. 5, no. 1, pp. 240–255, 2025.

[21] I. Sharafaldin, A. H., Lashkari, and A. A. Ghorbani,. "Toward generating a new intrusion detection dataset and intrusion traffic characterization," ICISSp, vol. 1, pp.108-116, 2018

[22] N. Moustafa, and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set) ," In 2015 military communications and information systems conference (MilCIS), pp. 1-6, 2015.

[23] Y. Yuan, W. Wang, and W. Pang, "A genetic algorithm with tree-structured mutation for hyperparameter optimisation of graph neural networks," In 2021 IEEE Congress on Evolutionary Computation, pp. 482-489, 2021.

[24] N. Zarrinpanjeh, F. D. Javan, H. Azadi, P. De Maeyer, and F. Witlox, "Ant colony optimization parameter selection for shortest path problem," In 24th International Society for Photogrammetry and Remote Sensing (ISPRS) , pp. 147-154, 2020.

[25] Luo, Y., Chen, R., Li, C., Yang, D., Tang, K. and Su, J., "An Improved Binary Simulated Annealing Algorithm and TPE-FL-LightGBM for Fast Network Intrusion Detection," Electronics, vol. 14, no. 2, p.231, 2025.

[26] B. Selvakumar, and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," Computers & Security, vol 81, pp.148-155, 2019.