Journal of Artificial Intelligence and Technology, 2026, 6, 815-829
https://doi.org/10.37965/jait.2026.1033

ISTP
RESEARCH ARTICLE

Securing IoT Networks Against Cyber Threats Using an Improved Whale Optimization Algorithm

Dhari M. Ali,[1] Mosleh M. Abualhaj,[1] Ahmad Abu-Shareha,[2] Mohamed Yousif,[3] Mohammad Sh. Daoud,[4] Muhammad Faheem,[5] and Hama Soltani[6]

[1]Department of Networks and Cybersecurity, Al-Ahliyya Amman University, Amman, Jordan
[2]Department of Data Science and Artificial Intelligence, Al-Ahliyya Amman University, Amman, Jordan
[3]School of Technologies, Cardiff Metropolitan University, Cardiff, UK
[4]College of Engineering, Al Ain University, Abu Dhabi, United Arab Emirates
[5]Faculty of Artificial Intelligence and Cyber Security, Universiti Teknikal Malaysia Melaka, 76100 Melaka, Malaysia
[6]Laboratory of Mathematics, Informatics and Systems (LAMIS), University of Tebessa, Tebessa, Algeria

(Received 27 November 2025; Revised 25 February 2026; Accepted 25 March 2026; Published online 22 September 2026)

Abstract: The advancement of Internet of Things (IoT) technology has enhanced efficiency and innovation, but it also presents new security challenges, particularly against sophisticated botnet cyberattacks. These attacks are too difficult for conventional security measures to handle, which emphasizes the need for better security solutions. This study presents a machine learning (ML) framework for securing IoT networks, using a modified whale optimization algorithm (WOA) for feature selection with an adaptive crossover function. The framework, tested on the N-BaIoT dataset, shows notable improvements in performance metrics across various classifiers for both binary and multiclass classification tasks. Particularly, classifiers such as random forest (RF), light gradient boosting machine (LGBM), adaptive boosting (AdaBoost), and histogram-based gradient boosting (HGB) achieved 100% accuracy, precision, and f1-score in binary classification. For multiclass classification tasks, the extreme gradient boosting (XGBoost), gradient boosting trees (GBTs), and artificial neural network (ANN) also demonstrated enhanced performance, with the XGBoost achieving 99.41% accuracy, precision, and f1-score.

Keywords: Botnet; feature selection; Internet of Things; machine learning; whale optimization algorithm

I. INTRODUCTION

Internet of Things (IoT) plays a significant role in daily life, from smart homes to automation in health care [1]. IoT improves convenience and overall quality of life. However, the surge in IoT adoption has attracted cybercriminals [2]. The number of IoT cyberattacks worldwide exceeded 112 million in 2022. Over recent years, this figure has increased significantly, from around 32 million cases detected in 2018 [3].

Cybercriminals frequently target IoT devices because traditional protection methods, such as firewalls and antivirus software, are inadequate [4]. In response to this challenge, machine learning (ML) has emerged as a vital tool for detecting cyber threats in IoT networks. Supervised learning, in particular, uses an ML algorithm trained on labeled data that includes both attacks and normal traffic collected from IoT networks. This process creates a framework that predicts whether data represent an attack or normal activity [5,6]. Figure 1 outlines the detection framework's role in IoT Networks. The framework is trained on captured IoT network traffic using a classification algorithm and then analyzes real-time traffic to flag it as suspicious (a cyberattack) or normal.

ML in network traffic analysis faces significant challenges because of the vast amounts of data involved, which can be computationally demanding [7]. Moreover, the presence of irrelevant data in this traffic may hinder the ML framework's performance in the detection process [8]. Therefore, feature selection is needed, a process that enhances the framework's performance by allowing it to focus only on relevant features for detection. It reduces computational demands by lowering the amount of data the framework needs to process [9,10].

There are various feature selection methods, including metaheuristic optimization algorithms. Metaheuristic algorithms are widely applied for solving complex problems across diverse domains such as medicine, engineering, and IT [11–13]. The whale optimization algorithm (WOA) is a metaheuristic algorithm known for its efficiency in these domains [14,15]. This study aims to use the WOA for feature selection in securing IoT networks and to improve its performance by incorporating a crossover function.

Crossover functions help algorithms avoid local optima, enabling more efficient exploration of the solution space by introducing new solutions. Local optimum is a common problem in optimization algorithms. It occurs when the algorithm reaches a solution that appears to be the best within the current solutions, but a better one could be found if the solution space were explored efficiently [16,17]. Therefore, this study seeks to build a robust ML framework to secure IoT networks by employing a modified WOA for feature selection on the N-BaIoT dataset.

Corresponding authors: Mosleh M. Abualhaj (e-mail: m.abualhaj@ammanu.edu.jo); Mohammad Sh. Daoud (e-mail: mohammad.daoud@aau.ac.ae).

© The Author(s) 2026. This is an open access article published under the CC BY license (https://creativecommons.org/licenses/by/4.0/).

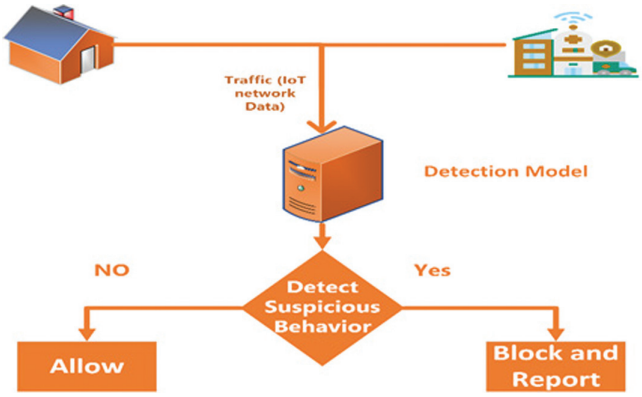


Fig. 1. Detection framework role in IoT networks.

IoT intrusion detection datasets are high-dimensional and contain redundant features, which increase computational cost and degrade detection performance in resource-constrained environments. The modified WOA algorithm is therefore motivated by the need to efficiently explore the feature space and select compact, informative subsets that reduce redundancy while maintaining detection accuracy.

II. BACKGROUND

This section provides an overview of the main concepts and components related to the proposed work. It discusses IoT network security challenges and the application of ML techniques, highlights the importance of feature selection in improving detection performance, introduces the WOA algorithm as a metaheuristic approach for feature selection, and describes the N-BaIoT dataset used in this study.

A. IoT NETWORK SECURITY AND ML

The growth of IoT devices and networks is accompanied by a rise in security challenges. The key challenges are summarized below [18–20]:

- **Limited Resources**: Many IoT devices lack sufficient storage or processing power (Imteaj *et al.*, 2021), and standard security measures consume too many resources for these devices (Kuzlu *et al.*, 2021).
- **Security With Scalability**: Large IoT networks may struggle to detect malicious traffic due to the high volume of data that flows through them (Khare and Totaro, 2019).
- **Securing Data Privacy and Integrity**: Protecting the data transmitted from the IoT devices over the internet is challenging; this data may include sensitive personal information (Khare and Totaro, 2019).
- **Network Accessibility**: IoT devices are often placed in public places where they can be accessed by anyone without permission. A compromised device could potentially put an entire network's security (Thakral *et al.*, 2022).
- **Life Cycle Management**: IoT devices often lack the capability for automatic updates and patches, rendering them exposed to unpatched security vulnerabilities (Tawalbeh *et al.*, 2020).

ML offers invaluable tools for security and improving the functionality of IoT networks (Al-Garadi *et al.*, 2020). In supervised learning, an ML framework is trained on datasets that contain both normal and attack data collected from IoT networks. After training the framework using an ML algorithm, it is implemented within an IoT network, serving as a security measure by recognizing and reporting network traffic that exhibits an attack pattern. This prevents threats in real time [21–23]. ML algorithms are an essential part of modern security, especially in securing IoT networks. Various ML algorithms can be used for securing IoT, including:

- **Random Forest (RF)** is an ensemble learning method that constructs multiple decision trees (DTs) during training. Each tree in the forest generates an output when a new input is introduced into the system. The outputs are then collected, and the most frequently seen output is selected as the final result. RF can handle missing values without reducing accuracy [24].
- **Light Gradient Boosting Machine (LGBM):** A fast gradient boosting framework based on DT algorithms, used for ranking, classification, and other ML tasks. It is designed to be distributed and efficient, with the following advantages: higher training speed and efficiency, lower memory usage, better accuracy, support for parallel and GPU learning, and the ability to handle large-scale data [25].
- **Adaptive Boosting (AdaBoost)**: An ensemble learning method that combines multiple weak learners to create a strong learner for improved accuracy. It adjusts the weights of incorrectly classified instances so that subsequent classifiers place greater emphasis on difficult cases [26].
- **Histogram-Based Gradient Boosting (HGB):** An efficient implementation of the gradient boosting framework, which uses histograms to bin continuous feature values and reduce the number of splitting points to consider, resulting in faster computation [27].
- **Artificial Neural Network (ANN):** A powerful computational framework inspired by the human brain, capable of learning from and interpreting data through a system of interconnected nodes mimicking biological neurons [28,29].
- **Extreme Gradient Boosting (XGBoost):** An optimized distributed gradient boosting library designed to be highly efficient, flexible, and portable. It implements ML algorithms within the gradient boosting framework and provides parallel tree boosting, solving many data science problems quickly and accurately [30].
- **Gradient Boosted Trees (GBTs):** This method builds the framework in a stage-wise fashion like AdaBoost, but it generalizes it by allowing optimization of an arbitrary differentiable loss function [31].

Each ML algorithm has its strengths and weaknesses, and the choice of algorithm often depends on the nature and volume of the data, as summarized in Table I.

B. FEATURE SELECTION IN IoT NETWORKS

Effective feature selection is crucial for optimizing ML frameworks, especially in complex domains such as IoT. By identifying and selecting only the most relevant data features for framework construction, performance is improved through increased accuracy and reduced computational costs [32–34]. This targeted approach can result in:

- **Improving performance**: Focusing on key traffic patterns enhances IoT security by improving detection accuracy and generalization across devices and attacks [34,35].

Table I. An overview of ML classifiers

Classifier	Type	Strengths	Weaknesses	Typical applications
RF	Ensemble (bagging-based)	Reduces overfitting, handles unbalanced data	Can be slow, prone to overfitting	Binary classification
ANN	Neural (deep learning)	Learns complex patterns	Requires a lot of data, prone to overfitting	Binary and multiclass classification, pattern recognition
LGBM	Ensemble (boosting-based)	Fast, handles large datasets, good with unbalanced data	May overfit on small datasets, complex parameter tuning	Binary and multiclass classification
AdaBoost		Combines weak learners, improves classification accuracy	Sensitive to noisy data and outliers, prone to overfitting	Binary classification, feature selection
HGB		Fast computation, efficient with large datasets	Can struggle with categorical features	Binary classification
XGBoost		High performance, handles missing data, regularization to avoid overfitting	Can be overcomplex and overfit, has time-consuming framework tuning	Multiclass classification, structured data
GBT		Sequential correction of predecessor's errors, robust	Requires careful tuning to avoid overfitting, slower training	Multiclass classification, regression tasks

- **Enhancing Computational Efficiency**: Feature elimination reduces data dimensionality and computational load, enabling faster training and real-time detection in resource-constrained IoT environments [34,35].

Multiple feature selection techniques exist, and among them, wrapper methods such as the WOA metaheuristic algorithm stand out. These techniques treat feature selection as a search problem, in which various feature sets are assembled and evaluated. They employ a predictive framework to evaluate each feature set, assigning it a score based on the framework's accuracy. In IoT, feature selection is important because of the vast amounts of data generated by devices and transmitted over the network. Moreover, the limited computational capacity of these devices makes it essential to reduce data dimensionality and processing load [34,36,37].

C. WOA METAHEURISTIC ALGORITHM

The WOA is a metaheuristic algorithm known for its efficiency in feature selection. The WOA simulates the social behavior and hunting strategy of humpback whales. The algorithm explores the solution space to identify the best solution by iteratively updating the positions of individual solutions (whales) while balancing exploration and exploitation [38,39].

It begins with the Initialization step, where potential solutions (whales) are created and essential parameters, such as the maximum number of iterations, are set. In the Iteration loop, the algorithm starts with the exploration and exploitation phase. In the exploration phase, the algorithm searches for better solutions. In the exploitation phase, it refines the current solutions. In both phases of exploration and exploitation in the loop, each whale's position is updated, and its fitness is evaluated. The Position Update involves two unique strategies, encircling prey and spiral updating positions. The algorithms end by returning the best solution found during all the iterations [39–41]. Figure 2 shows the main operations of the WOA algorithm.

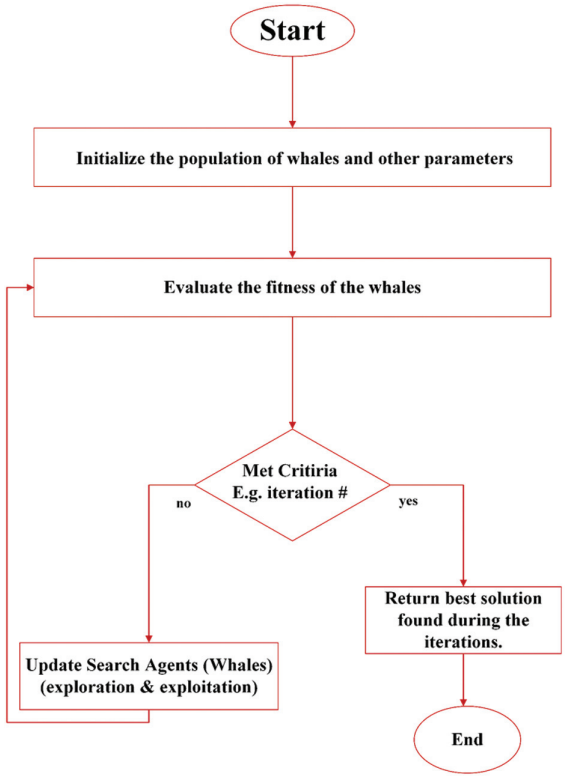


Fig. 2. Operations of the WOA algorithm.

D. N-BaIoT DATASET

The study used the N-BaIoT dataset, which contains network traffic data from both normal and compromised IoT devices. It contains 6,954,518 instances (records) and 115 features [42,43]. The dataset contains normal IoT traffic and attacks from Mirai and Bashlite, well-known botnets. Mirai is one of the most infamous botnets, known for its massive DDoS attacks. Mirai primarily targets IoT devices like IP cameras and home routers, exploiting weak security to create a large network of infected devices. Bashlite targets IoT devices intending to perform DDoS attacks. It exploits vulnerabilities in devices to gain control and coordinate large-scale network attacks [44–46].

The features in the N-BaIoT dataset are designed to capture a wide range of network data that can serve as indicators of normal or malicious behavior in IoT network traffic. N-BaIoT supports the creation of robust ML frameworks that can recognize and respond to a wide range of security issues for IoT networks. Table II lists the

Table II. Mirai and Bashlite botnet attack types

Attack type	Description
Mirai-scan	Automatic scanning for vulnerable devices
Mirai-ack	Ack flooding
Mirai-syn	Syn flooding
Mirai-UDP	UDP flooding
Mirai-UDPplain	UDP flooding with fewer options, optimized for higher packets per second
Bashlite-scan	Scanning the network for vulnerable devices
Bashlite-junk	Sending spam data
Bashlite-TCP	TCP flooding
Bashlite-UDP	UDP flooding
Bashlite-COMBO	Sending spam data and opening a connection to a specified IP address and port

attack types carried out by the Mirai and Bashlite botnets in the N-BaIoT dataset [42–44,46].

III. RELATED WORK

This section reviews studies that have applied the WOA to improve IoT network security, as well as those that use the N-BaIoT dataset to develop ML frameworks for IoT security.

A. WOA APPLICATIONS IN IoT SECURITY

The growing dependence on the IoT has heightened challenges related to the security and privacy of these systems. To address these issues, researchers have employed various approaches to secure IoT networks, including ML techniques and the WOA optimization algorithm. For example, Zhu and Zhu [47] introduced a framework for detecting anomalies in IoT-based networks using WOA. Their focus was on identifying irregularities that traditional firewall systems cannot detect. Specifically, they used an ensemble-based strategy, combining the classifiers support vector machine (SVM), K-nearest neighbors (KNN), and DT for classification, while WOA was applied for parameter tuning. The results show that the framework achieves 90.2% accuracy and 90% precision.

The study by Vijayanand and Devaraj [48] introduced an Intrusion Detection System (IDS) for Wireless Mesh Network (WMN), using the WOA with genetic crossover and mutation operators for feature selection, to enhance the SVM performance. The study's use of WOA with the genetic algorithm's crossover and mutation improved the original WOA in overcoming the premature convergence issue. The SVM with a Radial Basis Function (RBF) kernel function was used for classification on the CICIDS2017 and ADFA-LD datasets. The results showed that the SVM framework achieved precision scores of 95.91 on the CICIDS2017 dataset and 94.44 on the ADFA-LD dataset.

On the other hand, the study by Nagaraju *et al.* [49] proposed a deep learning (DL) framework for securing IoT Networks, utilizing WOA with Grey Wolf Optimizer (GWO) for feature selection, and Deep Convolutional Neural Networks (DCNN) for classification. The framework is assessed across four datasets (Google Code Jam, Leopard Mobile, Bot-IoT, and LITNET-2020). The results showed improvements in classification accuracy, specificity, sensitivity, and f1-score, with 99% accuracy on GCJ, 98% on Leopard Mobile, 99% on Bot-IoT, and 97% on LITNET-2020.

As proposed by Mafarja *et al.* [50], IoT networks were secured using WOA for feature selection, incorporating both V-shaped and S-shaped transfer functions. The study used the N-BaIoT dataset, a KNN classifier, and several innovative binarization rules. The results show that WOA, combined with specific transfer functions and binarization methods, outperforms other algorithms. The authors split the dataset into data1 {COMBO, UDP}, data2 {TCP, UDP}, data3 {SCAN, SYN}, data4 {UDP, ACK}, and data5 {TCP, UDPPLAIN}, reporting the highest accuracy as 99.99%, 99.78%, 99.41%, 99.52%, and 98.90%, respectively.

B. STUDIES UTILIZING THE N-BaIoT DATASET

A variety of studies have employed different IoT datasets. The following review focuses on research that specifically utilizes the N-BaIoT dataset. AZIMJONOV and Kim [51] implemented a stochastic gradient descent classifier (SGDC) using correlation coefficient-based algorithms for feature selection. The study was conducted on the datasets KDD-CUP-1999, BotIoT-2018, and N-BaIoT-2021. The SGDC classifier achieved an average accuracy of 92.69% across the datasets.

As for KASIM *et al.* [52], they introduced a hybrid framework to detect UDP flooding attacks in IoT networks, utilizing a method that combines Minimum Redundancy Maximum Relevance (MRMR) for feature selection with the Fine-tuned Stacked Sparse Autoencoder (FTSSAE) for classification on the N-BaIoT and NSL-KDD datasets. The FTSSAE classifier achieved 99.99% accuracy, precision, sensitivity, and specificity, as well as a 100% f1-score on the N-BaIoT dataset. As for the NSL-KDD, the results were 97.81% accuracy, 97.53% precision, 99.89% sensitivity, 87.83% specificity, and 98.70% for the f1-score.

The study by Palla and Tayeb [53] introduces an approach to detecting Mirai malware on IoT devices using ML algorithms. The study's novel contribution lies in the implementation and comparison of ANN and RF frameworks in MATLAB and Python, respectively. Using the "N-BaIoT" dataset, which comprises real traffic data from nine IoT devices, including the Damini Doorbell, each with 115 features, the ANN framework achieves 67.5% accuracy and f1-scores.

The research by Khoa *et al.* [54] introduces a cooperative learning framework in which these filters trained on deep neural network (DNNs) leverage data from their local networks to train cyberattack detection systems. These frameworks are then shared among IoT gateways, leveraging the collective intelligence to enhance the overall accuracy and speed of intrusion detection while reducing network traffic and protecting data privacy across the system. The system's efficacy is validated through simulations using real datasets, including the N-BaIoT dataset, which contains normal and malicious traffic from nine IoT devices, with each dataset comprising benign traffic and 10 attack types from Mirai and Bashlite. The proposed approach demonstrates a substantial improvement in classification accuracy (up to 14.76%) and a 98.5% reduction in communication overhead compared to conventional ML methods. For the Danmini Doorbell, the system achieved an accuracy of 99.84%.

The study by Gatea and Hameed [55] introduced an IoT botnet detection framework based on regression and linear discrimination analysis. This innovative approach leverages multiple linear regression (MLR) for feature selection, streamlining the detection process by focusing on the most relevant characteristics of botnet traffic. Using the N-BaIoT dataset, the framework achieves exceptional accuracy, demonstrating its effectiveness compared to conventional

methods. The framework results for the Danmini Doorbell device were 99.82% (no feature selection), 99.80% (75% features), 99.68% (50% features), and 86.20% (25% features).

Unlike existing WOA variants and other feature selection approaches, the proposed method introduces an adaptive fitness-guided crossover mechanism that dynamically biases offspring generation toward higher-quality solutions. This design improves exploration while maintaining low computational overhead, effectively reducing feature redundancy without relying on additional hybrid optimizers or complex binarization strategies. As a result, the proposed approach is particularly well suited for high-dimensional IoT intrusion detection data, where efficient search, compact feature subsets, and scalability are critical.

IV. PROPOSED METHODOLOGY

This study aims to develop an ML framework designed to enhance the security of IoT networks. It will use the N-BaIoT dataset, focusing on data from the Damini Doorbell device, to train and evaluate the framework. However, before the framework is trained, the essential preprocessing steps are performed. The preprocessing steps are transformation, normalization, and feature selection. The feature selection stage uses the modified WOA algorithm, enhanced with a crossover function to efficiently explore the solution space (feature subsets) and find a better subset that improves the framework's performance. The outcome of the preprocessing stage is the use of inputs for various ML algorithms to identify the most effective classifier for improving the security of IoT networks. Figure 3 shows the stages for the ML framework development and evaluation.

To prevent data leakage, the dataset was first partitioned into training and testing subsets. All preprocessing steps, including feature selection, normalization, and model training, were performed exclusively on the training data. The learned transformations were then applied to the test set, which was not used at any stage of model construction or parameter tuning.

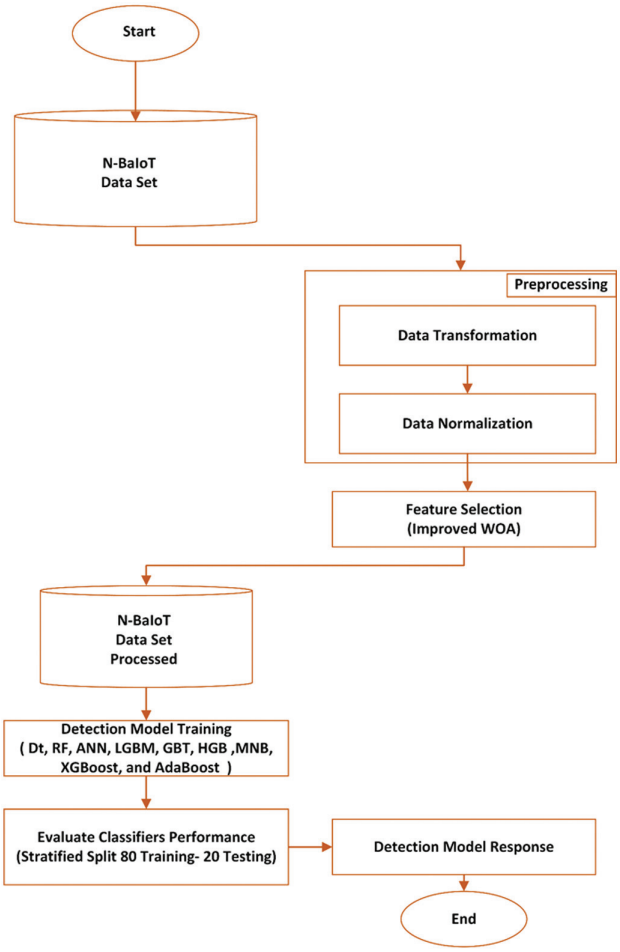


Fig. 3. Proposed ML-based IoT attack detection model.

Attack Label	Encoded Label
benign	0
gafgyt_combo	1
gafgyt_junk	2
gafgyt_scan	3
gafgyt_tcp	4
gafgyt_udp	5
mirai_ack	6
mirai_scan	7
mirai_syn	8
mirai_udp	9
mirai_udpplain	10

Fig. 4. Encoded class labels used in the N-BaIoT dataset.

A. DATA TRANSFORMATION

Data transformation is a necessary step in preparing data for the proposed ML framework (Dogan and Birant, 2021). It is implemented as a label encoder that converts data labels to numerical values. Because ML algorithms tend to work better with numerical data, converting label values to a more suitable numerical representation is a fundamental step in optimizing the ML framework. The transformation is done using label encoding. Label encoding is a well-established data transformation technique that assigns a unique integer to each distinct category in a feature [45]. Figure 4 shows the encoded labels for the N-BaIoT dataset.

In binary classification, benign traffic is labeled 0, and all other attack types are labeled 1. In multiclass classification, benign traffic is labeled as 0. The attacks are divided into families, with each family representing a different type of attack behavior. For example, attacks from the Bashlite family are numerically encoded from 1 to 5, corresponding to various attack methods such as combo, junk, scan, tcp, and udp. Similarly, different types of Mirai attack methods are encoded as 6-10, corresponding to ack, scan, syn, udp, and UDPplain, respectively.

B. DATA NORMALIZATION

Data normalization is an essential step in preparing data for a framework (Alam and Mohanty, 2022). Normalization adjusts variables to a common scale, and the MinMaxScaler is a common and effective method that rescales features to a range of 0 to 1 (Kappal, 2019). This process is crucial because classification algorithms and the WOA assume that all input variables are on the same scale; otherwise, they may be adversely affected. Therefore, the MinMaxScaler is implemented to ensure all features are on the

same scale. The MinMaxScaler is calculated using Equation (1) [45,56]:

$$\mathbf{X_{Scaled}} = \frac{\mathbf{X} - \mathbf{X_{Min}}}{\mathbf{X_{Max}} - \mathbf{X_{Min}}} \quad (1)$$

where X_Scaled is the result for the MinMaxScaler, X is the original value, X_Min is the minimum value in the column, and X_Max is the maximum value in the column. The MinMaxScaler was used for the features in the dataset.

C. FEATURE SELECTION USING THE PROPOSED MODIFIED WOA ALGORITHM

This section outlines the proposed modification of the WOA algorithm for feature selection in IoT security, introducing an adaptive crossover function to enhance exploration and avoid local optima.

1). PROPOSED MODIFIED WOA ALGORITHM. The WOA algorithm is known for its proficiency in solving optimization problems and enhancing feature selection. However, it is not immune to convergence on local optima, which can limit its effectiveness. Local optima refer to reaching an optimal solution from the selected candidate solutions, but not the optimal solution for the entire solution space [16,17]. Therefore, to mitigate this issue and enhance the WOA's ability to explore the solution space, an adaptive crossover function is introduced to the algorithm, thereby improving its search capabilities and avoiding local optima. Algorithm 1 outlines the pseudo-code for the proposed modified WOA.

Algorithm 1 shows the steps for the modified WOA with the modification outlined in blue, and all these steps are described in the following:

- **Step (1):** Set the number of solutions to explore (N) and how many times to loop (T).
- **Step (2):** Randomly generate N potential solutions, known as whales.
- **Step (3):** Loop until the number of iterations (t) reaches the limit (T).
- **Step (4):** Start the main loop until the stopping criteria (iteration t < T) are met.
- **Step (4.1):** Evaluate the quality of each WOA using a fitness function.
- **Step (4.2):** Identify the best solution X(best) at the current iteration.

Algorithm 1. The Pseudo-Code for the Modified WOA

1. Initialize the population size N and the maximum number of iterations T.
2. Initialize N whales X_i (i = 1, 2, . . . , N) using random initialization, where each X_i represents a potential solution.
3. While (iteration t < T #stopping Criteria is not met) do
   4.1. Calculate the fitness values of all whales using a fitness function.
   4.2. Identify the best solution found so far as X(best).
   4.3. For each whale (X_i) do
      4.3.1. Update the position of Xi based on the best solution found so far X(best), and the positions of other whales in the population.
      4.3.2. Apply the crossover function
5. Return $X_{(best)}$

- **Step (4.3):** Iterate through every whale.
- **Step (4.3.1):** Change the whales' positions guided by the best current solution and influenced by the positions of other whales.
- **Step (4.3.2):** Create new solutions using the crossover function by combining different whales, then add them to the population while maintaining the population size N by keeping the better solutions and removing the less fit ones.
- **Step (5):** Return the best feature subset X(best).

2). ADAPTIVE CROSSOVER FUNCTION FOR ENHANCED EXPLORATION. Introducing an adaptive crossover function into the WOA can significantly boost the algorithm's ability to explore the solution space when securing IoT networks. The crossover function helps the WOA in avoiding local optima. The adaptive crossover function is determined by the blending factor alpha, which is calculated from the fitness of the parent solutions using Equation 2:

$$\text{alpha} = \text{fitness2}/(\text{fitness1} + \text{fitness2}) \quad (2)$$

where if fitness1 and fitness2 are close, alpha will be around 0.5, meaning both parents contribute almost equally to the offspring; if one parent has significantly higher fitness, alpha will lean more toward that parent. In short, adaptively in the crossover process is achieved by adapting the fitness value of the parents producing the offspring. It selects more features from the parent with the higher fitness value to produce feature subsets that inherit the parent's good characteristics, thereby leading to better solutions. Unlike traditional crossover mechanisms that use fixed or random mixing ratios, the proposed crossover is adaptive because the blending factor alpha is dynamically adjusted based on the fitness of the parent solutions. This fitness-guided adaptively biases offspring generation toward higher-quality feature subsets, improving exploration efficiency and accelerating convergence while reducing redundant features. Figure 5 shows a representation of the crossover process when the blending factor alpha is 0.5 (i.e., the fitness of parent 1 equals the fitness of parent 2).

The adaptive crossover process comprises three steps:

- **Step 1**: Randomly select pairs of solutions (whales) from the population, excluding the top two based on fitness. Implement the adaptive crossover between each pair. The adaptive aspect comes from the blending factor alpha, which is calculated based on the fitness of the parent solutions. This ensures that

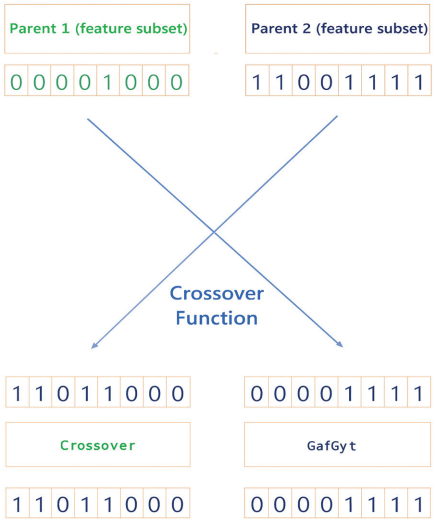


Fig. 5. Representation of the crossover process.

more of the new offspring's characteristics are inherited from the parent with the higher fitness. This step introduces two new offspring. These offspring combine attributes from both parents, enriching the population's diversity with new characteristics that may lead to better solutions.

- **Step 2**: Identify the two most fit solutions in the population and apply the adaptive crossover to them to create two additional offspring. These offspring will inherit the most successful features of their parents, potentially leading to faster, more efficient convergence toward the optimal solution.
- **Step 3**: The algorithm selects the top N solutions (N is the original population size) based on their fitness to maintain a stable population size. With the introduction of new offspring, the increase in population size ensures that the most fit solutions are retained for future iterations, whereas the less effective solutions are discarded.

In short, the modified WOA incorporates a crossover function to generate new feature subsets, enhancing exploration and population diversity. This improvement is particularly effective in reducing redundant features in high-dimensional IoT traffic data while converging toward compact feature subsets with lower computational cost.

D. CLASSIFICATION

Classification is important for securing IoT networks, predicting the value of network data, and determining whether traffic is benign or an attack. Many classification algorithms were used to train the framework because each classifier has a unique method for identifying patterns and making predictions. Some classifiers might work better with IoT network data. Therefore, utilizing a variety of classifiers increases the likelihood that a framework will be discovered that performs well for securing IoT networks. The used classification algorithms are RF, ANN, LGBM, AdaBoost, HGB, XGBoost, and GBT [24–31].

The dataset employed in this study was partitioned using a stratified sampling approach with an 80:20 training-to-testing ratio to preserve class distribution across subsets. A total of 68,750 instances were utilized, comprising 34,375 samples from each class, thereby ensuring a balanced representation of both categories. This stratified and balanced sampling strategy enhances the reliability and validity of the framework's performance evaluation by mitigating potential bias arising from class imbalance.

V. IMPLEMENTATION AND RESULTS

This section explores the implementation environment, implementation stages, evaluation metrics, detailed results for the modified WOA, and a comparative analysis between the modified WOA framework and other frameworks.

A. IMPLEMENTATION

The research was conducted on Windows 10, and Python was chosen for its extensive support for ML libraries. The implementation and evaluation of the ML framework requires a high-performance computing setup. Some of the libraries used in implementing the ML framework include pandas, numpy, sklearn.framework_selection, math, sklearn.preprocessing, and matplotlib.pyplot.

The proposed framework was implemented through several sequential stages. First, data labels were transformed using a label encoder. Next, data normalization was performed using MinMaxScaler. Afterward, feature selection was performed using the proposed modified WOA algorithm. Finally, binary and multiclass classification was completed.

- In binary classification, the algorithms RF, ANN, LGBM, AdaBoost, and HGB were used to classify IoT network traffic as either benign or attack traffic. These algorithms have demonstrated effectiveness in binary classification [57,58].
- In multiclass classification, ANN, XGBoost, and GBT were selected for their ability to handle multiple output classes and complex relationships, making them suitable for classifying IoT network traffic as benign or one of 10 attack types in the datasets. These algorithms excel in multiclass data scenarios [59,60].

B. IMPLEMENTATION OF THE PROPOSED MODIFIED WOA

The parameters for the modified WOA were set to a population size of 20 and a maximum of 200 iterations. The population size was set to 20 because the crossover function doubles the population size with each offspring, making it sufficient to explore potential solutions during each iteration and thus allowing for better exploration of the WOA. Using 200 iterations gives the algorithm plenty of opportunities to refine solutions over time, increasing the likelihood of reaching the optimal solution and thus allowing better exploitation of the WOA.

The fitness function used DT to calculate the error rate, considering both prediction accuracy and the number of selected features, which is shown in Equation (3), where a lower fitness value indicates a more optimal feature subset:

$$\text{Fitness function} = \text{alpha} * (\text{1-Accuracy}) + \text{beta} * \left(\frac{\text{number of selected features}}{\text{maximum number of features}}\right) \quad (3)$$

The weighting coefficients are fixed as alpha = 0.9 and beta = 0.1 to emphasize classification accuracy while still penalizing large feature subsets, reflecting the importance of detection performance in security applications. During the feature selection process, the fitness value is computed exclusively using the training data, whereas the test set is strictly reserved for the final performance evaluation. This evaluation protocol ensures a fair assessment and prevents overfitting or bias toward the test split.

The fitness function incorporates two objectives, maximizing classification accuracy while penalizing the selection of a large number of features. The inner-validation is conducted by examining various feature subsets generated by the modified WOA. Then, during testing, the optimal feature subset identified in the training phase is evaluated on a separate portion of the dataset. This portion serves as both validation and testing sets, as will be detailed in the results section.

1). FEATURE SELECTION RESULT FOR BINARY CLASSIFICATION. The convergence plot in Fig. 6 shows the relationship between the fitness of the best solution found at each iteration and the modified WOA across 200 iterations for binary classification. It shows the algorithm converges to an optimal solution after the 10th iteration.

The modified WOA selected 10 key features out of 115. These features are (MI_dir_L5_weight, H_L3_variance, HH_L1_weight, HH_L0.01_pcc, HH_jit_L3_mean, HpHp_L5_std, HpHp_L5_pcc,

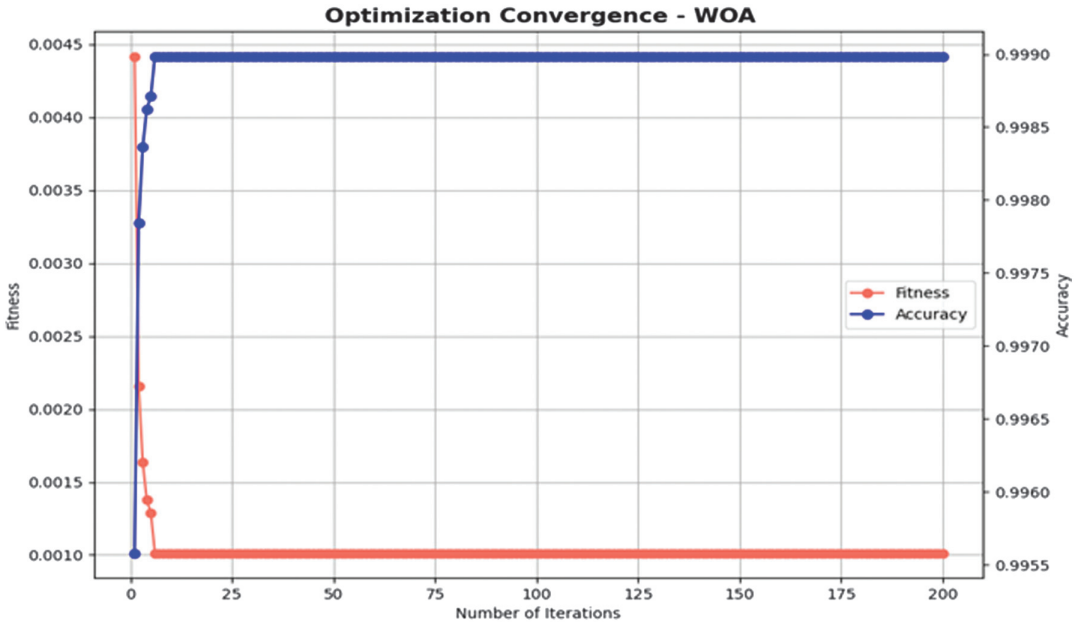


Fig. 6. Convergence plot of the WOA (binary classification).

HpHp_L3_mean, HpHp_L1_covariance, and HpHp_L0.1_covariance). The importance of the selected features was determined using permutation importance. Permutation importance measures the influence of individual features on the framework's accuracy, indicating each feature's importance. Figure 7 shows the importance of the selected features; it shows that only five features have an impact on the framework's accuracy, with the features (HH_jit_L3_mean, MI_dir_L5_weight) being the most influential.

The selected features emphasize global coordination and deviation from normal IoT traffic behavior. Mutual information and entropy features indicate structured and unstable attack traffic, whereas jitter, host-based, and hypothesis-based features capture synchronized and anomalous behavior, enabling reliable separation of benign and malicious traffic.

2). FEATURE SELECTION RESULT FOR MULTICLASS CLASSIFICATION. The fitness of the best solution at each iteration is shown in Fig. 8, the convergence plot for feature selection using the modified WOA in multiclass classification. It shows the correlation between the fitness of the best solution found at each iteration and the iteration number. The plot demonstrates the algorithm's stability, with an optimal solution reached after the 30th iteration.

Eight features were chosen from the modified WOA in multiclass classification. These features are (MI_dir_L0.1_variance, MI_dir_L0.01_mean, H_L3_weight, HH_L1_pcc, HH_jit_L5_variance, HH_jit_L1_mean, HpHp_L3_std, and HpHp_L1_radius).

Permutation importance was used to determine the importance of the chosen features. It assesses the impact of individual features on the framework's accuracy and identifies the importance of each. Figure 9 illustrates the importance of the chosen features, indicating that only four features have a noticeable impact on the framework's accuracy, with the largest impact for the feature MI_dir_L0.01_mean in multiclass classification.

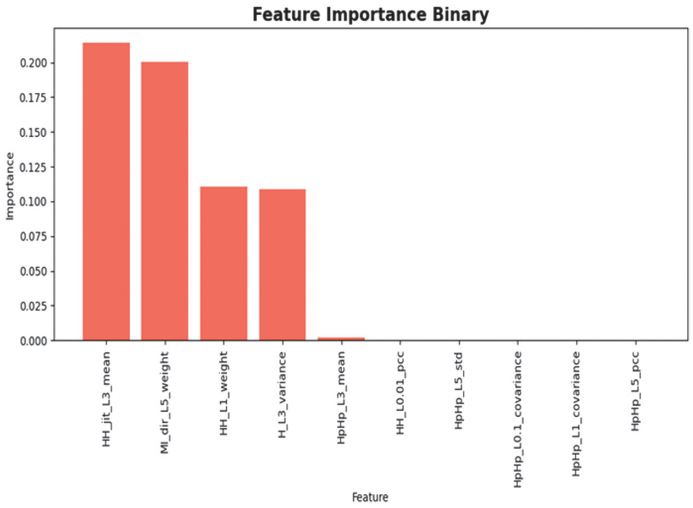


Fig. 7. Importance of the selected features (binary classification).

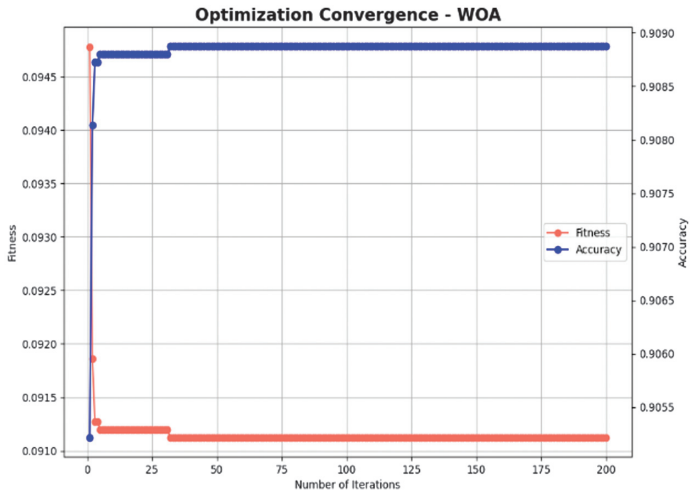


Fig. 8. Convergence plot of the WOA (multiclass classification).

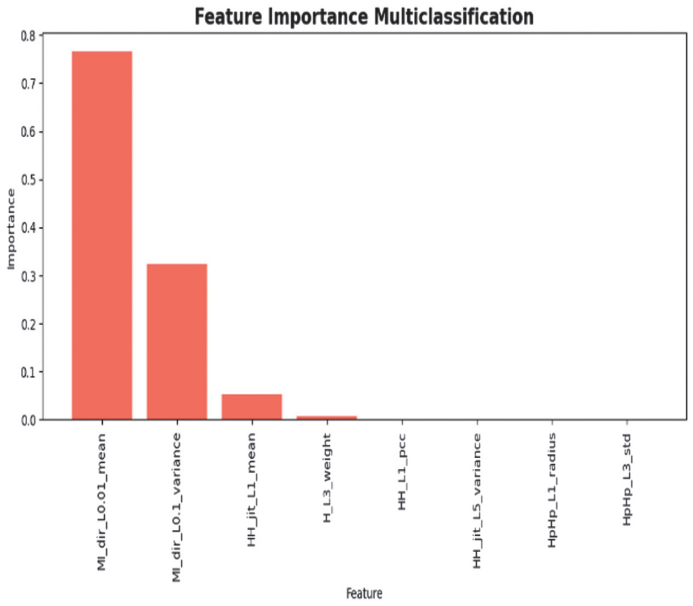


Fig. 9. Importance of the selected features (multiclass classification).

The selected features capture temporal dependency, host synchronization, and traffic instability across network layers, which are critical for distinguishing different botnet attack types. Mutual information, entropy, and jitter features reflect coordinated and bursty traffic patterns, while host-based and hypothesis-driven features highlight abnormal correlations and deviations characteristic of Mirai and Bashlite attacks.

C. EVALUATION METRICS

Various metrics were used to evaluate the framework's performance in detecting threats in IoT networks, and these metrics and their formulas are [61–64]:

- **True Positive (TP)**: Correctly predicted attack traffic.
- **True Negative (TN)**: Correctly predicted benign traffic.
- **False Negative (FN)**: Malicious traffic misclassified as benign.
- **False Positive (FP)**: Benign traffic misclassified as malicious.
- **Accuracy**: As seen in Equation (4), accuracy is the ratio of correct detections to total detections N. It measures the model's overall correctness:

$$\textbf{Accuracy} = \frac{(\textbf{TP} + \textbf{TN})}{\boldsymbol{N}} \tag{4}$$

- **Precision**: As seen in Equation (5), it is the ratio of the correct TP to the total of predicted positives. It evaluates the model's performance regarding true positives:

$$\textbf{Precision} = \frac{(\textbf{TP})}{(\boldsymbol{TP} + \boldsymbol{FP})} \tag{5}$$

- **F1-score**: As seen in Equation (6), it is the balance of precision and recall:

$$\textbf{F1-score} \frac{2 * (\boldsymbol{Precision} * \boldsymbol{Recall})}{(\boldsymbol{Precision} + \boldsymbol{Recall})} \tag{6}$$

- **Convergence Time**: Speed of the training process.

Evaluating the framework across these diverse metrics ensures a thorough understanding of its capability to correctly predict malicious and benign IoT network traffic.

D. RESULTS AND DISCUSSION

The results are shown in the three sections that follow. The results for the modified WOA framework are displayed in Section V.A. The performances of the original and modified WOA frameworks are compared in Section V.B. Also, a comparative analysis between the modified WOA framework and other frameworks appears in Section V.C.

1). THE MODIFIED WOA FRAMEWORK RESULTS. In the evaluation of classifiers for binary classification of IoT network traffic, the classifiers RF, LGBM, AdaBoost, and HGB all demonstrated impeccable performance, with each scoring 100% across the accuracy, precision, and f1-score metrics. Regarding convergence time, LGBM was the fastest among these at 0.179 seconds, while RF was the slowest at 2.009 seconds. Additionally, among all evaluated models, ANN was the slowest at 20.013 seconds. The ANN classifier, while slightly worse, achieved a high accuracy, precision, and f1-score of 99.83% but had a considerably longer convergence time, slower than the other classification methods. Figures 10–13 show the accuracy, precision, f1-score, and convergence time for the proposed modified WOA algorithm.

Multiclass classification results are presented in Fig. 14–17. Notably, XGBoost stands out with exceptional scores—over 99.41% in accuracy, precision, and f1-score—and a reasonable convergence time. In comparison, GBT achieves respectable scores, particularly in precision, but requires a much longer convergence time. Although ANN is slower and less accurate than XGBoost and GBT, it still demonstrates decent classification capabilities. Specifically, Fig. 14 shows the ranking of classifiers by accuracy, with XGBoost leading at 99.41%, followed by GBT at 90.6% and ANN at 84.67%. Figure 15 then ranks the classifiers by precision in multiclass classification of IoT network traffic, while Fig. 16 highlights XGBoost's superior performance in balancing the f1-score. Finally, Fig. 17 displays convergence times, with XGBoost converging the fastest and GBT the slowest. In summary, the classifiers display varying effectiveness in multiclass classification tasks, with XGBoost proving most proficient overall, GBT demonstrating slower convergence but good precision, and ANN offering a trade-off between speed and accuracy.

2). THE MODIFIED WOA vs. THE ORIGINAL WOA. In binary classification, the modified WOA improved the accuracy of ML

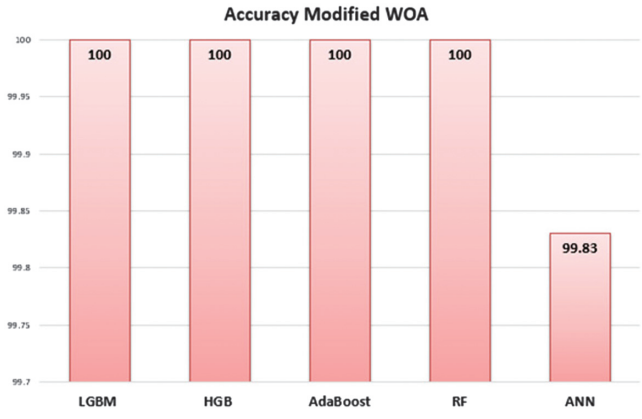


Fig. 10. Ranking of classifiers in terms of accuracy (binary classification).

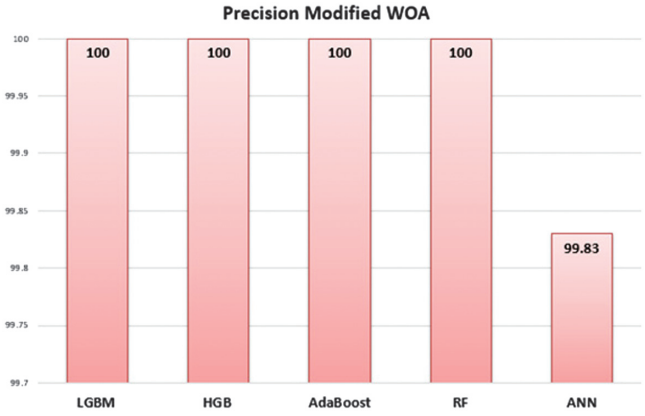


Fig. 11. Ranking of classifiers in terms of precision (binary classification).

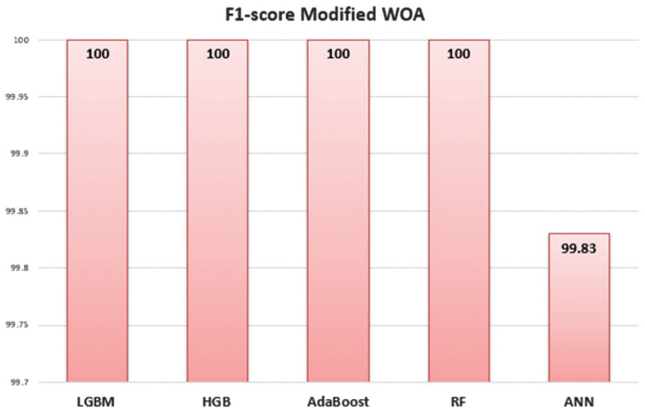


Fig. 12. Ranking of classifier in terms of f1-score (binary classification).

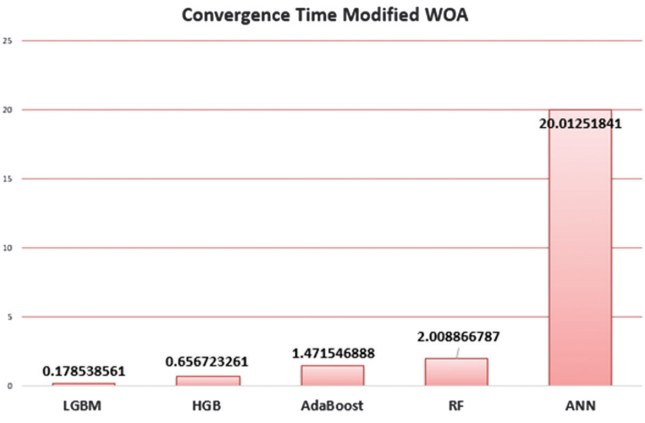


Fig. 13. Ranking of classifiers in terms of convergence time (binary classification).

algorithms for IoT network traffic. The RF, AdaBoost, LGBM, and HGB accuracy rates improved from 99.99%, 99.99%, 99.96%, and 99.91%, respectively, using the original WOA, to 100% each using the modified WOA. Also, the ANN's accuracy increased from 89.6% to 99.83%, an impressive 11.42% gain. Figure 18 outlines the improvements in accuracy using the modified WOA vs. the original WOA.

The modified WOA has led to notable improvements in precision across several ML algorithms for binary classification

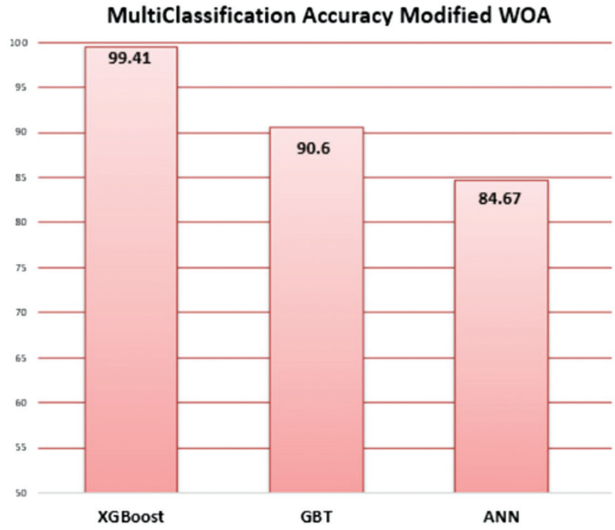


Fig. 14. Ranking of classifiers in terms of accuracy (multiclass classification).

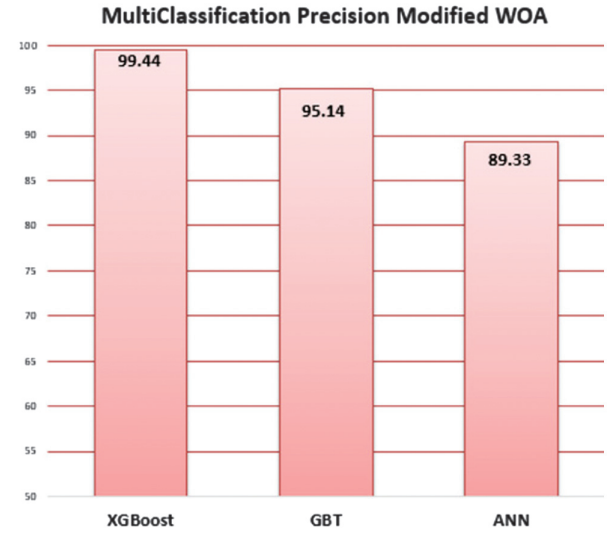


Fig. 15. Ranking of classifiers in terms of precision (multiclass classification).

of IoT network traffic. The RF algorithm's precision is now perfect at 100%, up from an already high 99.99% in the original WOA. The ANN's precision also jumped from 91.4% to 99.83%. LGBM and AdaBoost, like RF, now achieve ideal 100% precision. HGB joins these algorithms with perfected precision, improving from 99.91% to 100%. Figure 19 shows these precision improvements across various ML algorithms using the proposed WOA.

The improvements are not limited to precision but are also reflected in the accuracy and f1-score metrics, where each of these ML algorithms has achieved 100% accuracy with the modified WOA process. Figure 20 shows the improvements in f1-score for the various ML algorithms using the proposed modified WOA.

In multiclass classification, the modified WOA improves multiple ML classifiers. XGBoost shows small gains: accuracy from 99.33% to 99.41%, precision from 99.37% to 99.44%, and f1-score from 99.33% to 99.41%. GBT also improves slightly, with accuracy rising from 90.26% to 90.6%, precision from 94.85% to 95.14%, and f1-score from 87.23% to 87.58%. By contrast, the ANN

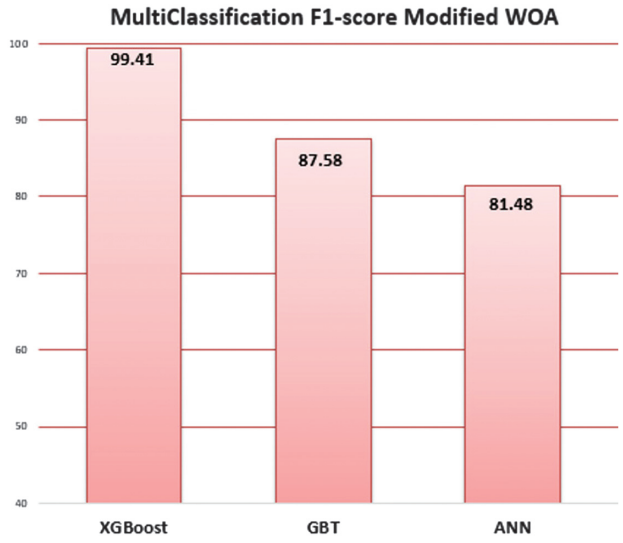


Fig. 16. Ranking of classifiers in terms of f1-score (multiclass classification).

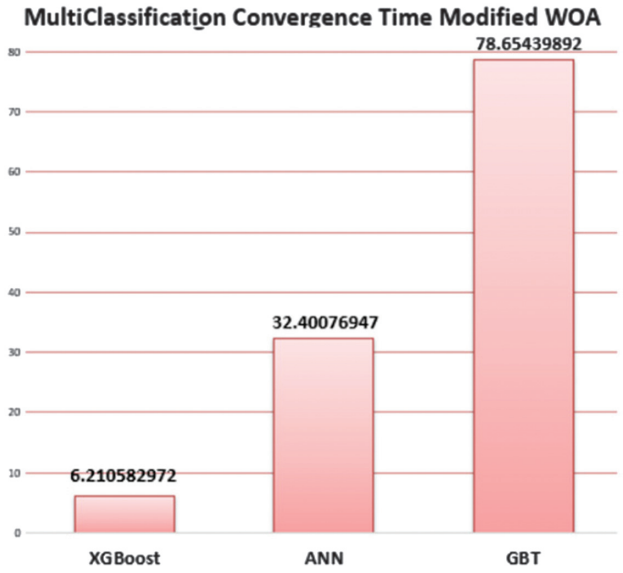


Fig. 17. Ranking of classifiers in terms of convergence time (multiclass classification).

classifier sees larger improvements: accuracy increases from 76.84% to 84.67%, precision from 80.1% to 89.33%, and f1-score from 72.25% to 81.48%. Figures 21–23 highlight the modified WOA's influence on ML classifier performance for detecting malicious IoT traffic, measured by accuracy, precision, and f1-score.

These results show that the modified WOA has led to better classification outcomes in terms of accuracy, precision, and f1-score.

3). THE MODIFIED WOA vs. OTHER FRAMEWORKS. Using the N-BaIoT dataset (Danmini Doorbell device), the proposed modified WOA framework was evaluated and compared with several existing approaches reported in the literature. The comparative results, summarized in Table III, clearly demonstrate the proposed framework's superior accuracy.

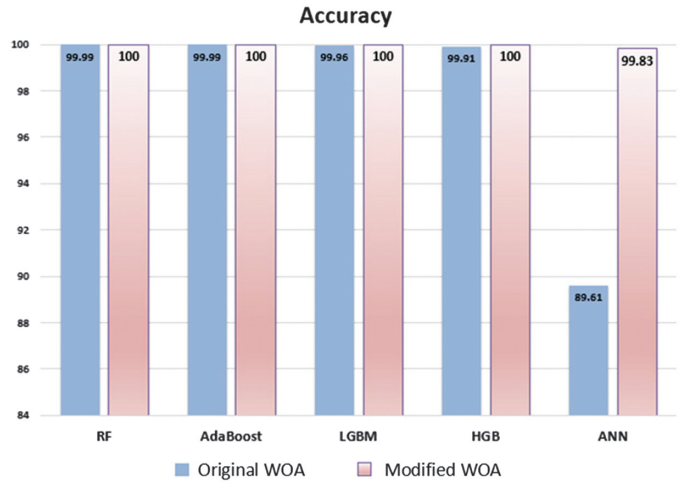


Fig. 18. Comparison of accuracy between the modified and original WOA (binary classification).

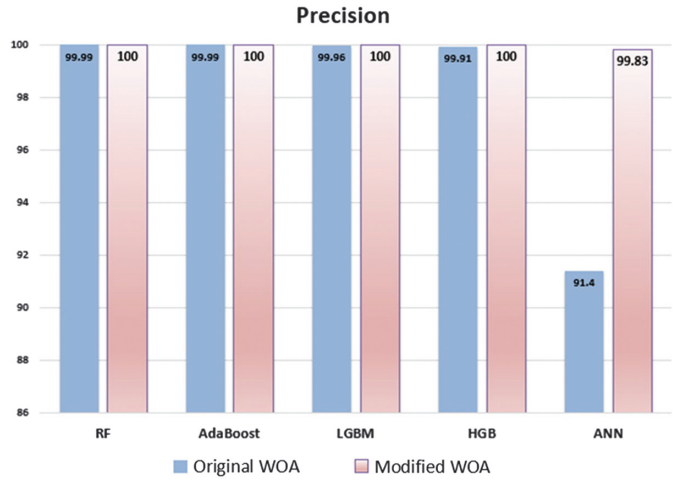


Fig. 19. Comparison of precision between the modified and original WOA (binary classification).

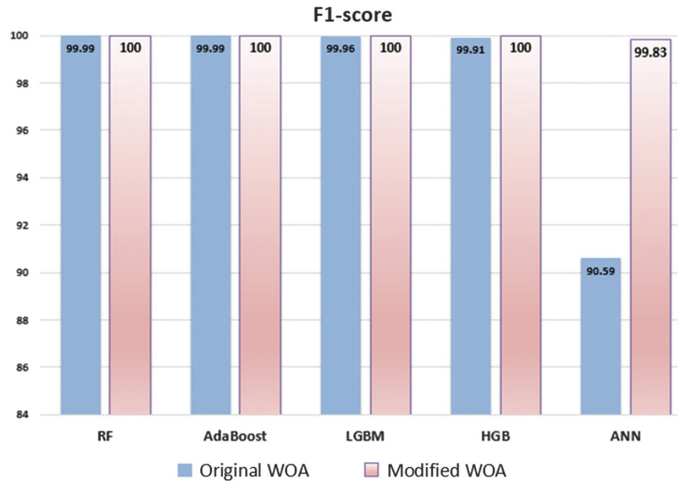


Fig. 20. Comparison of f1-score between the modified and original WOA (binary classification).

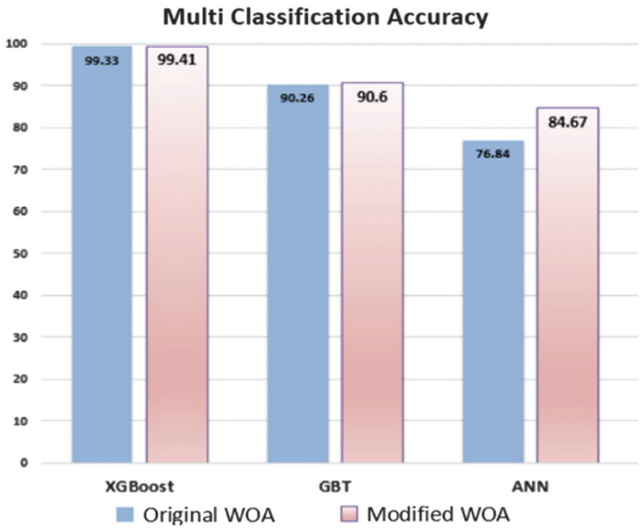


Fig. 21. Comparison of accuracy between the modified and original WOA (multiclass classification).

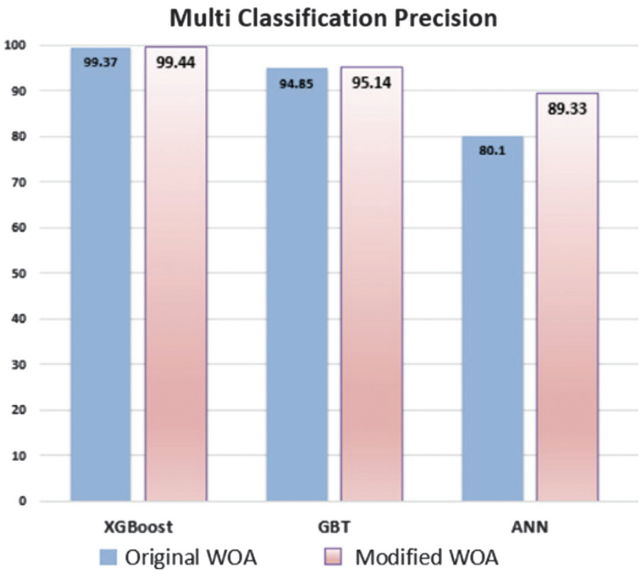


Fig. 22. Comparison of precision between the modified and original WOA (multiclass classification).

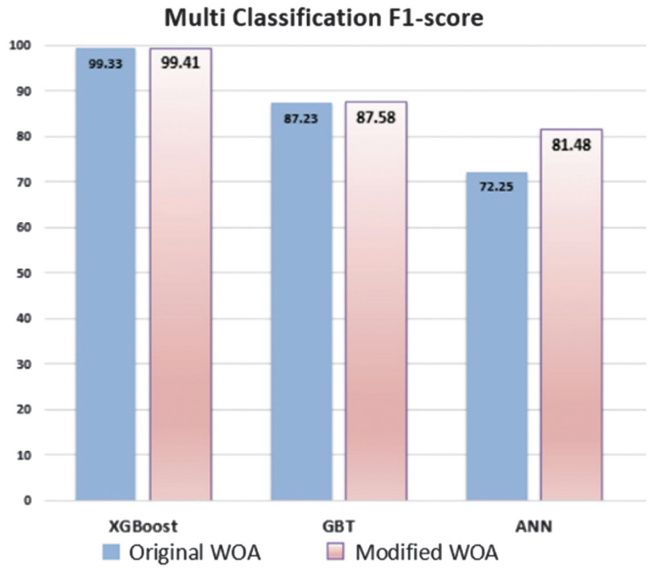


Fig. 23. Comparison of f1-score between the modified and original WOA (multiclass classification).

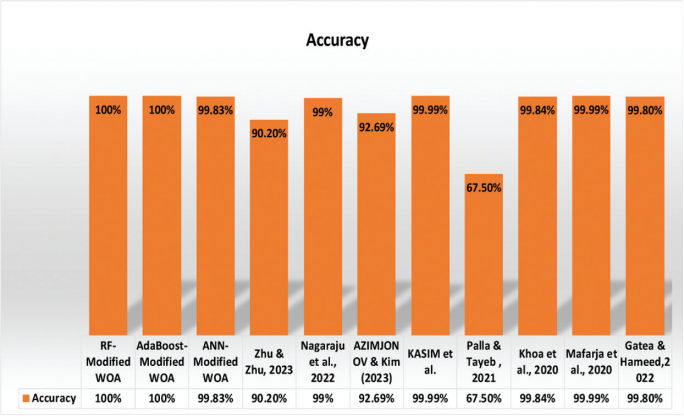


Fig. 24. Accuracy comparison between existing frameworks and the modified WOA.

The RF-modified WOA and AdaBoost-modified WOA models achieved 100% accuracy, confirming their ability to perfectly distinguish between benign and attack instances within the dataset. The ANN-modified WOA model closely followed, achieving 99.83% accuracy, reflecting its strong generalization capability and stability. In comparison, previous studies reported significantly lower results, such as Zhu and Zhu (2023) with 90.20% accuracy and Palla and Tayeb (2021) with 67.50% accuracy, which emphasizes the notable improvement achieved by the proposed approach.

Moreover, other studies such as Nagaraju *et al.* (2022), AZIMJONOV and Kim (2023), and Khoa *et al.* (2020) achieved accuracies of 99%, 92.69%, and 99.84%, respectively—still slightly below the performance of the proposed framework. Similarly, KASIM *et al.* and Mafarja *et al.* (2020) achieved very high

Table III. Comparative performance of the proposed framework and existing methods

Method	Accuracy (%)	Precision (%)
RF-modified WOA	100	100
AdaBoost-modified WOA	100	100
ANN-modified WOA	99.83	99.83
Zhu and Zhu, 2023	90.20	90
Nagaraju *et al.*, 2022	99	NA
AZIMJONOV and Kim (2023)	92.69	NA
KASIM *et al.*	99.99	100
Palla and Tayeb, 2021	67.50	NA
Khoa *et al.*, 2020	99.84	NA
Mafarja *et al.*, 2020	99.99	NA
Gatea and Hameed,2022	99.80	NA
Vijayanand and Devaraj, 2020	NA	95.91

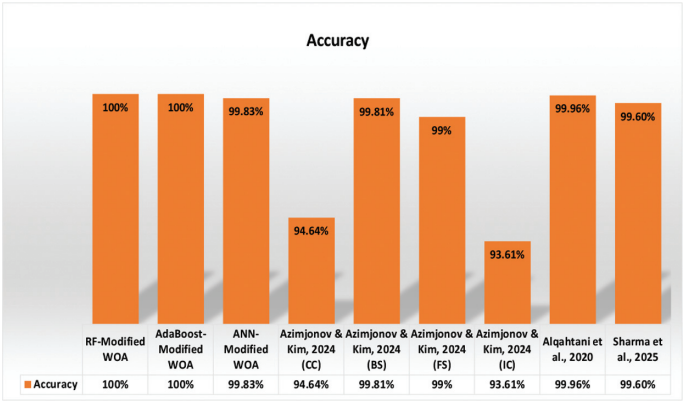


Fig. 25. Accuracy comparison between widely used baselines and the modified WOA.

accuracies of 99.99%, comparable to the proposed modified WOA model. Meanwhile, Gatea and Hameed (2022) reported 99.80% accuracy, which, although competitive, remains marginally lower than the best-performing modified WOA-based classifiers.

Overall, the modified WOA framework consistently matches or surpasses the results of state-of-the-art models, reaffirming its robustness and efficiency in feature selection and classification for IoT-based intrusion detection using the N-BaIoT Danmini Doorbell dataset. Figure 24 presents a comparative illustration of the accuracy performance between the proposed modified WOA framework and other related models.

4). THE MODIFIED WOA vs. WIDELY USED BASELINES. The proposed modified WOA framework was compared with widely used feature selection baselines on the N-BaIoT dataset, as illustrated in Fig. 25. The RF- modified WOA and AdaBoost-modified WOA models achieved 100% accuracy, while the ANN-modified WOA model achieved 99.83% accuracy, demonstrating consistent and robust performance across different classifiers. In contrast, Azimjonov and Kim [65] employed correlation coefficient (CC), backward sequential (BS), forward sequential (FS), and importance coefficient (IC) feature selection methods, reporting accuracies of 94.64%, 99.81%, 99%, and 93.61%, respectively. Alqahtani *et al.* [66] used Fisher Score (FS) and achieved 99.96% accuracy, while Sharma *et al.* [67] applied random forest-based feature ranking (RF-Rank), reporting 99.6% accuracy. Overall, the modified WOA consistently outperforms these widely used baseline feature selection approaches by achieving higher classification accuracy. This result highlights the effectiveness of the adaptive, fitness-guided search mechanism in selecting compact and highly discriminative feature subsets for IoT intrusion detection.

VI. LIMITATIONS, SCOPE, AND FUTURE WORK

Despite the strong experimental results, this study is conducted under controlled conditions using an offline, supervised learning setting and traffic collected from a single IoT device (Danmini Doorbell) within the N-BaIoT dataset, which includes Mirai and Bashlite botnet attacks. Consequently, the reported performance reflects dataset-specific characteristics and may not fully capture the variability, heterogeneity, and dynamics encountered in real-world IoT deployments. In particular, the observed perfect performance in some classification tasks should be interpreted within this controlled context and not assumed to directly generalize to all operational environments. Furthermore, the use of a stratified random split, while appropriate for controlled benchmarking, may yield optimistic performance estimates if temporally adjacent or behaviorally correlated traffic samples appear across training and testing partitions.

Although precautions were taken to prevent data leakage—such as performing feature selection and fitness evaluation exclusively on the training data—there remains a potential risk of overfitting and data distribution bias due to the homogeneous nature of the dataset and the use of a single device. Traffic patterns in real IoT environments may differ over time or across devices, and such distribution shifts could impact model performance. In addition, the absence of explicit temporal stamps, session identifiers, or multi-device grouping metadata in the utilized subset limits the ability to implement stricter evaluation protocols such as time-aware, session/group-wise, or cross-device splits in a fully reproducible manner within this study. Therefore, the current evaluation primarily demonstrates the discriminative capability of the proposed feature selection framework rather than guaranteed robustness under all deployment conditions.

The primary contribution of this work lies in the proposed optimization-based feature selection framework using the modified WOA, rather than in claiming universal generalization across heterogeneous IoT devices, traffic patterns, or attack families. The experimental evaluation is designed to provide a clear benchmark of the framework's effectiveness under the stated assumptions. Performance may vary when different datasets, classifiers, network conditions, or evolving attack behaviors are considered.

Future work will extend the evaluation to more realistic and diverse scenarios, including multi-device settings, cross-device and cross-botnet generalization analysis, and time-aware or sequential training and testing strategies to mitigate overfitting risks and assess robustness under distribution shifts. Such stricter evaluation protocols are expected to provide a more conservative and deployment-oriented estimate of model generalization. In addition, incorporating online or adaptive learning mechanisms and evaluating the framework on additional IoT datasets will further strengthen result credibility and practical applicability.

VII. CONCLUSION

The rapid growth of IoT technology has improved connectivity and automation but also introduced serious security challenges. This study proposed an ML-based framework to enhance IoT network security using a modified WOA integrated with an adaptive crossover function for feature selection. The modified WOA showed superior exploration capability and effectively avoided local optima, resulting in the selection of highly relevant and discriminative features. Using the N-BaIoT dataset, the proposed framework achieved outstanding performance across multiple ML classifiers. In binary classification, RF, LGBM, AdaBoost, and HGB achieved perfect scores of 100% in accuracy, precision, and f1-score. For multiclass classification, XGBoost achieved 99.41% accuracy, precision, and f1-score, while the adaptive crossover improved results across all classifiers—most notably enhancing ANN accuracy by 11% compared to the original WOA. These results confirm the effectiveness of the modified WOA-based framework in improving IoT attack detection and overall classification performance.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no conflicts of interest to report regarding the present study.

REFERENCES

[1] A. Sharma *et al.*, "Advancements in 5G-IoT technology-based healthcare: A focus on healthcare monitoring for smart cities," in *Mobile Crowdsensing and Remote Sensing in Smart Cities*, C. A. Kerrache, Y. Sahraoui, C. T. Calafate and A. M. Vegni, Eds. Cham: Springer Nature Switzerland, 2024, pp. 73–84.

[2] T. M. Ghazal *et al.*, "Data space privacy model with federated learning technique for securing IoT communications in autonomous marine vehicles," *J. Intell. Rob. Syst.*, vol. 111, no. 3, p. 84, 2025.

[3] A. Petrosyan, "Worldwide-annual-internet-of-things-attacks," 2023, May 3. Available from statista: https://www.statista.com/statistics/1377569/worldwide-annual-internet-of-things-attacks/.

[4] A. R. Shorman *et al.*, "Adaptive intrusion detection for iot networks using artificial immune system techniques: A comparative study," *J. Robot. Control (JRC)*, vol. 6, no. 2, pp. 570–582, 2025.

[5] M. M. Abualhaj *et al.*, "Enhanced network communication security through hybrid dragonfly-bat feature selection for intrusion detection," *J. Commun.*, vol. 20, no. 5, pp. 607–618, 2025.

[6] K. P. Sharma *et al.*, "Hybrid convolutional neural network for robust attack detection in wireless sensor networks," *Internet Technol. Lett.*, vol. 8, no. 6, p. e650, 2025.

[7] M. Shehab, A. Smerat, and L. Abualigah, "22 reinforcement learning-based optimization algorithms: A survey," in *Mastering the Minds of Machines: A Journey into Deep Learning and AI*, L. Abualigah, Ed. Boca Raton, FL: CRC Press, 2025, pp. 172–177.

[8] Q. Shambour, M. Al-Zyoud, and O. Almomani, "Quantum-inspired hybrid metaheuristic feature selection with SHAP for optimized and explainable spam detection," *Symmetry*, vol. 17, no. 10, p. 1716, 2025.

[9] Y. Sanjalawe *et al.*, "Smart load balancing in cloud computing: Integrating feature selection with advanced deep learning models," *PLOS One*, vol. 20, no. 9, pp. 1–50, Sep. 2025, doi: 10.1371/journal.pone.0329765.

[10] S. Daoud *et al.*, "A novel deep learning-based spider wasp optimization approach for enhancing brain tumor detection and physical therapy prediction," *Comput. Methods Programs Biomed. Update*, vol. 7, p. 100193, 2025.

[11] R. Masadeh *et al.*, "Narwhal optimizer: A nature-inspired optimization algorithm for solving complex optimization problems," *Comput. Mater. Contin.*, vol. 85, no. 2, pp. 3709–3737, 2025.

[12] M. Braik *et al.*, "Advancements in global optimization with an empowered capuchin search algorithm," *Neural Comput. Appl.*, vol. 37, pp. 1–43, 2025.

[13] A Chaotic Boost: The Chaotic Crayfish Optimization Algorithm for Superior Solution Quality.

[14] M. M. Abualhaj *et al.*, "Comparative analysis of whale and Harris Hawks optimization for feature selection in intrusion detection," *Indones. J. Electr. Eng. Comput. Sci.*, vol. 37, no. 1, pp.179–185, 2025.

[15] P. Nandal *et al.*, "Improving underwater image quality through real-ESRGAN with whale optimization algorithm," *Internet Technol. Lett.*, vol. 8, no. 4, p. e70047, 2025.

[16] R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Android malware detection using a modified dwarf mongoose algorithm," *Int. J. Intell. Eng. Syst.*, vol. 18, no. 8, Art. no. ijies2025.0930.21, 2025, doi: 10.22266/ijies2025.0930.21.

[17] R. Alabdallat, M. Abualhaj, and A. Abu-Shareha, "Enhanced Multi-class Android Malware Detection Using a Modified Dwarf Mongoose Algorithm", *Int. J. Anal. Appl.*, vol. 23, pp. 1–23, Jan. 2025, doi: https://doi.org/10.28924/2291-8639-23-2025-0.

[18] Sensorless real-time solar irradiance prediction in grid-connected PV systems using PSO-MPPT and IoT-enabled monitoring.

[19] M. Z. Iskandarani, "Effect of intelligent reflecting surface on WSN communication with access points configuration," *IEEE Access*, vol. 13, pp. 13380–13394, 2025.

[20] M. Z. Iskandarani and K. Chakrabarti, "Adaptive frequency and energy modeling framework for robotic wireless sensor networks based on path loss predictions," *J. Rob.*, vol. 2025, no. 1, p. 6156608, 2025.

[21] An IoT-Based Active Learning Convolutional Neural Networks Model for Predicting River Water Quality.

[22] The Intersection of AI and the Internet of Things (IoT): Transforming Data into Intelligence.

[23] A. M. Ali, A. Al-Qerem, and H. A. Owida, "Assessing WLAN 802.11 ac vs. 802.11 n using OPNET: An evaluation of comparative performance," in Future of Information and Communication Conference, K. Arai, Eds. Cham: Springer Nature Switzerland, March 2025, pp. 656–670.

[24] M. Alkhawatrah, "Random Forests relay selector in buffer-aided cooperative networks," *IEEE Access*, 2025.

[25] Y. Wang *et al.*, "Low-complexity fast CU classification decision method based on LGBM classifier," *Electronics*, vol. 12, no. 11, p. 2488, 2023.

[26] Y. Ding *et al.*, "An efficient AdaBoost algorithm with the multiple thresholds classification," *Appl. Sci.*, vol. 12, no. 12, p. 5872, 2022.

[27] P. Theerthagiri, "Liver disease classification using histogram-based gradient boosting classification tree with feature selection algorithm," *Biomed. Signal Process. Control*, vol. 100, p. 107102, 2025.

[28] J. A. Solairaju *et al.*, "Development and validation of ANN models for water absorption in sawdust and kenaf fiber-reinforced polystyrene composites," *Eng. Rep*, vol. 7, no. 9, p. e70393, 2025.

[29] N. Ayub *et al.*, "A decision framework for intra task fixed priority INTEL PXA270 distributed architecture for soft RT-applications based on deep learning," *Eng. Technol. Appl. Sci. Res*, vol. 15, no. 3, pp. 23553–23558, 2025.

[30] S. Khalid *et al.*, "Tab transformer with meta-ensemble learning approaches for enhanced diabetes prediction," *PeerJ Comput. Sci.*, vol. 11, p. e3206, 2025.

[31] M. Douiba *et al.*, "An improved anomaly detection model for IoT security using decision tree and gradient boosting," *J. Supercomput.*, vol. 79, no. 3, pp. 3392–3411, 2023.

[32] Z. Awais *et al.*, "ISCC: Intelligent Semantic Caching and Control for NDN-enabled industrial iot networks," *IEEE Access*, vol. 13, pp. 169881–169898, 2025.

[33] A. M. Ali *et al.*, "Traffic monitoring and control system for smart city pollution regulation using IoT and correlated capsule networks," *IEEE Access*, 2025.

[34] L. Abualigah *et al.*, "Optimizing intrusion detection in wireless sensor networks via the improved Chameleon Swarm Algorithm for feature selection," *IET Commun.*, vol. 19, no. 1, p. e70029, 2025.

[35] S. N. Makhadmeh *et al.*, "A crossover-integrated Marine Predator Algorithm for feature selection in intrusion detection systems within IoT environments," *Internet Things*, vol. 31, p. 101536, 2025.

[36] M. J. Alenazi *et al.*, "Deep reinforcement learning based flow aware-QoS provisioning in SD-IoT for precision agriculture," *Comput. Intell.*, vol. 41, no. 1, p. e70023, 2025.

[37] S. Mali *et al.*, "Federated reinforcement learning-based dynamic resource allocation and task scheduling in edge for IoT applications," *Sensors (Basel)*, vol. 25, no. 7, p. 2197, 2025.

[38] M. M. Abualhaj *et al.*, "Enhancing intrusion detection system performance using a hybrid of Harris Hawks and Whale Optimization Algorithms," *Eng. Technol. Appl. Sci. Res*, vol. 15, no. 4, pp. 24354–24361, 2025.

[39] G. Kaur *et al.*, An efficient approach for localizing sensor nodes in 2D wireless sensor networks using whale optimization-based naked Mole Rat Algorithm. Mathematics, 2024.

[40] M. M. Abualhaj *et al.*, "Performance comparison of Whale and Harris Hawks optimizers with network intrusion prevention systems," *J. Appl. Data Sci.*, vol. 5, no. 4, pp. 1530–1538, 2024.

[41] N. M. Allayla, F. Nazar Ibraheem, and R. Adnan Jaleel, "Enabling image optimisation and artificial intelligence technologies for better Internet of Things framework to predict COVID," *IET Netw.*, pp. 1–11, 2022.

[42] M. Hossain *et al.*, "Machine learning approaches for detecting iot botnet attacks: A comparative study of n-baiot dataset," in *2024 International Conference on Decision Aid Sciences and Applications (DASA)*. Manama, Bahrain: IEEE, Dec. 2024, pp. 1–7.

[43] C. Okur, A. Orman, and M. Dener, "DDOS intrusion detection with machine learning models: N-BaIoT data set," in *The International Conference on Artificial Intelligence and Applied Mathematics in Engineering*. D. Jude Hemanth, T. Yigit, U. Kose and U. Guvenc, Eds. Cham: Springer International Publishing, May 2022, pp. 607–619.

[44] F. Yousaf *et al.*, "Machine learning-based detection of Mirai and Bashlite Botnets in IoT networks," *J. Comput. Biomed Inf.*, vol. 7, no. 01, pp. 678–689, 2024.

[45] A. Mosleh *et al.*, "A paradigm for DoS attack disclosure using machine learning techniques," *Int. J. Adv Comput. Sci. Appl.*, vol. 13, no. 3, 2022.

[46] S. Barrett, B. Boswell, and G. Dorai, "Exploring the vulnerabilities of IoT devices: A comprehensive analysis of mirai and bashlite attack vectors," in *2023 10th International Conference on Internet of Things: Systems, Management and Security (IOTSMS)*, M. Quwaider, and Y. Jararweh, Eds. San Antonio, TX: IEEE, Oct. 2023, pp. 125–132.

[47] Z. Zhu and M. Zhu, "A novel method for anomaly detection in the Internet of Things using Whale Optimization Algorithm," *Int. J. Adv Comput. Sci. Appl.*, vol. 14, no. 5, 2023.

[48] R. Vijayanand and D. Devaraj, "A novel feature selection method using whale optimization algorithm and genetic operators for intrusion detection system in wireless mesh network," *IEEE Access*, vol. 8, pp. 56847–56854, 2020.

[49] R. Nagaraju *et al.*, "Attack prevention in IoT through hybrid optimization mechanism and deep learning framework," *Meas.: Sens.*, vol. 24, p. 100431, 2022.

[50] M. Mafarja *et al.*, "Augmented whale feature selection for IoT attacks: Structure, analysis and applications," *Future Gener. Comput. Syst.*, vol. 112, pp. 18–40, 2020.

[51] J. Azimjonov and T. Kim, "Stochastic gradient descent classifier-based lightweight intrusion detection systems using the most efficient feature subsets of datasets," Available:SSRN 4378339.

[52] Ö. KASIM, Deeper Fine-Tuned Autoencoder for User Datagram Protocol Flooding Network Traffic Detection in Internet of Things, 2023.

[53] T. G. Palla and S. Tayeb, "Intelligent Mirai malware detection for IoT nodes," *Electronics*, vol. 10, no. 11, p. 1241, 2021.

[54] T. V. Khoa *et al.*, "Collaborative learning model for cyberattack detection systems in IoT industry 4.0," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. Seoul, Korea (South): IEEE, May 2020, pp. 1–6.

[55] M. J. Gatea and S. M. Hameed, "An Internet of Things Botnet detection model using regression analysis and linear discrimination analysis," *Iraqi J. Sci.*, pp. 4534–4546, 2022.

[56] Y. A. Maz *et al.*, "Transfer learning-based approach with an ensemble classifier for detecting Keylogging attack on the Internet of Things," *Comput. Mater. Contin.*, vol. 85, no. 3, 2025.

[57] M. Shahhosseini and G. Hu, "Improved weighted random forest for classification problems," in *Progress in Intelligent Decision Science. IDS 2020. Advances in Intelligent Systems and Computing, vol 1301*, T. Allahviranloo, S. Salahshour, and N. Arica, Eds. Springer, Cham: Springer International Publishing, 2021, pp. 42–56. doi: 10.1007/978-3-030-66501-2_4

[58] A. Sharma and B. Singh, "AE-LGBM: Sequence-based novel approach to detect interacting protein pairs via ensemble of autoencoder and LightGBM," *Comput. Biol. Med.*, vol. 125, p. 103964, 2020.

[59] P. M. Abhilash and D. Chakradhar, "Prediction and analysis of process failures by ANN classification during wire-EDM of Inconel 718," *Adv. Manuf.*, vol. 8, pp. 519–536, 2020.

[60] T. T. H. Le, Y. E. Oktian, and H. Kim, "XGBoost for imbalanced multiclass classification-based industrial internet of things intrusion detection systems," *Sustainability*, vol. 14, no. 14, p. 8707, 2022.

[61] M. M. Abualhaj *et al.*, "Spam detection boosted by firefly-based feature selection and optimized," *Int. J. Adv. Soft Comput. Appl.*, vol. 17, no. 3, pp. 1–19, Nov. 2025.

[62] M. M. Abualhaj *et al.*, "Smart firewall for phishing detection powered by bio-inspired algorithms," *J. Adv. Inf. Technol.*, vol. 16, no. 11, pp. 1529–1539, 2025.

[63] A. A. Abu-Shareha *et al.*, "Diabetes prediction using hybrid supervised and unsupervised techniques based on PIMA dataset. *J. Artif. Intell. Technol.*, vol. 6, pp. 79–87, 2025.

[64] O. Almomani *et al.*, Adaptive Enhanced Grey Wolf Optimizer for Efficient Cluster Head Selection and Network Lifetime Maximization in Wireless Sensor Networks, 2025.

[65] J. Azimjonov and T. Kim, "Designing accurate lightweight intrusion detection systems for IoT networks using fine-tuned linear SVM and feature selectors," *Comput. Secur.*, vol. 137, p. 103598, 2024.

[66] M. Alqahtani, H. Mathkour, and M. M. Ben Ismail, "IoT botnet attack detection based on optimized extreme gradient boosting and feature selection," *Sensors*, vol. 20, no. 21, p. 6336, 2020.

[67] K. P. Sharma *et al.*, "Interpretable intrusion detection for IoT environments using a self-attention-based explainable AI framework," *Sci. Rep.*, vol. 15, no. 1, p. 39937, 2025.