Intrusion Detection in IoT Networks Using Extra Trees, Random Forests, and Hybrid Optimization Algorithms

Xiaonan Chen
Wuxi Vocational College of Science and Technology, Wuxi, Jiangsu, 214028, China
E-mail: wxcxn@126.com

Keywords: machine learning, extra trees, internet of things, random forest, intrusive attacks.

Received: December 13, 2024

The Internet of Things (IoT) is essentially the physical objects that communicate over the internet with the purpose of remote measurement and control. In the area of IoT network security, it is crucial that all types of attacks on these networks are correctly recognized and that intelligent intrusion detection is done with high accuracy and at high speed. Thus, a model has been proposed in this paper to identify the intrusion of the IoT network with the help of optimized Machine Learning (ML) models. The core classifiers used here are Extra Trees (EXT) and Random Forest (RF). In the effort of enhancing the correctness of prediction, 6 various optimization algorithms, such as Grey Wolf Optimizer (GWO), Hunger Games Search (HGS), Moth-Flame Optimization (MFO), Satin Bowerbird Optimization (SBO), Slime Mould Algorithm (SMA), and Whale Optimization Algorithm (WOA), were employed for the hyperparameter (HPs) tuning of the main classifiers. The authors performed their evaluation tests on the CICIoT2023 dataset that contains over 1.19 million labeled network traffic records with 47 features and 33 different attack types. These were captured in a smart home-like IoT topology of 105 devices. The performance criteria were Accuracy, Precision, Recall, and F1-score. The EXT-MFO hybrid model from the proposed set registered the highest performance by an accuracy of 90.72% and an F1-score of 0.905. These results exceeded those of other EXT-based and RF-based variants. Finally, the accuracy of different models was also checked by performing a case study on the dataset that contains 10 cyberattacks in an IoT topology with 105 devices. The results indicated that the EXT-MFO hybrid model outperformed other models in terms of F1-score values and, therefore, was more accurate. The findings reveal that the most suitable optimizers for energy optimization of EXT and RF classifiers are the MFO and SMA algorithms, respectively. The findings also suggested that, generally, the models on the EXT classifier complemented by different optimizers are the ones that use more time for HP optimization. The results show that the EXT-MFO model outperforms all other State-of-the-Art (SOTA) optimization algorithms in the Internet of Things (IoT) Intrusion Detection (ID). Therefore, the EXT-MFO hybrid model, which is based on the research results, is put forward to find out the attempted attacks aimed at the IoT network.

Povzetek: Članek obravnava zaznavanje napadov v IoT omrežjih z modeloma Extra Trees in Random Forest, katerih hiperparametre optimizirajo algoritmi Grey Wolf, Hunger Games Search, Moth-Flame, Satin Bowerbird, Slime Mould in Whale Optimization. Najbolje se izkaže kombinacija Extra Trees + Moth-Flame.

1 Introduction

In the world of technology and communication, the IoT is a new concept [1]. The IoT, first presented by Kevin Ashton in 1999, refers to the connection between various physical devices and objects through the Internet all over the world. Through the IoT, individuals and objects can be linked to anything and anyone at any moment, in any location, and preferably through any channel, network, and resource. The crucial aim of IoT is to improve human existence and make it more controllable by combining tangible tools and digital understanding [2], [3]. The term IoT refers to the connection of objects in the surrounding environment to the Internet network. IoT provides ease and quality of life for consumers with different technologies. The IoT is made up of intricate devices that exchange significant quantities of data automatically, and the quantity of these devices grows on a daily basis [4]. As device usage continues to grow, more users have become susceptible to cyber-attacks. Therefore, the existing threats against these devices must be analyzed in order to create protective mechanisms against them. Since the IoT relies on the Internet to function, security should be considered its most important challenge [5].

Today, the total number of IoT devices reaches billions. These numbers store massive data warehouses. Most of this data includes people's personal information, such as contact information, environmental information, and health measurements. An attack occurs when intruders attempt to infiltrate a device or data stream. Attack scenarios can include stealing or modifying data, hijacking devices or entire systems, and connecting devices to other virtual networks. If an IoT system is compromised, it will negatively impact the system's operation and could lead to complete failure. The data in the system is mostly real-time and has a great impact on the owners and their businesses [6], [7]. Cybersecurity is crucial in today's information-rich world, especially with the IoT. Devices connected to IoT networks are exposed to cyber-attacks, and their security is one of the main concerns that affect their usability. Existing methods for securing IoT applications need to be improved in terms of real-time data analysis and prevention of cyber-attacks [8]. Instant detection improves network security, as it speeds up warning and disconnection of affected IoT devices from the network, thus preventing the spread of botnets and further outbound attack traffic. Numerous investigations have been performed in this area up to this point to accurately and immediately identify attacks on the IoT network. Although this research has been able to enhance the accuracy of ID, it still has challenges [9], [10]. However, most of these approaches suffer from restricted capacity to adjust to new kinds of attack, high False Positive (FP) rates, and insufficient response speed for real-time detection. Therefore, the network remains vulnerable, especially in dynamic IoT environments. The IoT is currently one of the newest research topics. The devices that benefit from this technology send information and receive commands, which is why there is a possibility of hackers intruding while sending and receiving information. There are many security challenges in the field of IoT, and solving these challenges is very important, as it lets users be sure that devices equipped with IoT are safe from any vulnerability. As the global deployment of IoT devices rapidly expands and the traffic volume of IoT-based on the increasing Distributed Denial of Service (DDoS) attacks, it is crucial to identify such attacks in time to reduce the risks associated with them [11], [12], [13]. IoT is defined as a heterogeneous technology that provides new services and innovations in different application areas. Categorizing attacks in the field of IoT and placing them in categories based on the type and method of attack makes it easier to identify different attacks on IoT and ways to deal with them [14]. Although IoT devices include various smart devices and systems connected to the Internet that serve household and organizational needs, they usually lack strict security protocols. Attackers have used this fact to infiltrate an organization's system through any of the vulnerabilities of IoT devices [15].

An IoT attack is any unwanted and malicious activity that targets devices linked to the IoT. These attacks are mainly carried out to increase unauthorized control, manipulate or steal information, destroy devices, or even use the device as a tool for attacks on networks or other systems [16]. In IoT attacks, attackers may use Internet-connected devices to send many requests to a service or device to damage or occupy resources. These attacks may interrupt communications or reduce device performance [17]. IoT attacks occur in various forms and affect IoT devices, related networks, or even the communication services of these devices. In the following, some examples of IoT attacks are introduced: 1- Intrusion attacks, in which the attacker attempts to penetrate the IoT device and take control of it. This penetration is done through security weaknesses in the software or hardware of the device. 2- Denial of Service (DoS) attack, in which the attacker attempts to saturate the resources of the IoT device by generating much traffic or sending unauthorized requests in order to interrupt the service or reduce its efficiency. 3- Man-in-the-middle attack (MitM), which is an example of an IoT attack where the intruder interferes with the connection of the IoT device with the server or another device, and not only changes the data that is sent, but also inserts the desired information. 4- Firmware Attacks that are open to vulnerabilities if the IoT device has no software update, and also if the attacker's intention is so, they can get the device under their control by changing or adding code to the device's firmware. 5- An attacker uses a network of IoT devices as a botnet to perform attacks on other networks or services, which are known as Thingbot Attacks. 6- Security Exploits, that is, the case when the perpetrator is trying to sneak into the system using the software or hardware weaknesses of IoT devices. To counter IoT attacks, it is more beneficial for manufacturers and users to adopt necessary security measures, namely, software upgrading, encryption, Attack Detection (AD) and response, and effective security policy development [18], [19], [20], [21].

The following section discusses numerous studies relevant to the research topic. Chen et al. (2020) suggested a multi-layered AD system to prevent DDoS attacks in the IoT, and by implementing the Decision Tree (DT) algorithm on the Data Sensor data set, the accuracy indices and 1F score reached more than 97% [22]. Hussain et al. (2020) introduced an approach for transforming network data into a 3-channel picture using a Residual Network (ResNet), which is one of the Deep Learning (DL) models. Also, the 2019 CICDDOS dataset was used to detect attacks. Afterward, Convolutional Neural Networks (CNN), such as the ResNet network, were trained using the altered data from this dataset because of their notable effectiveness in image processing. Using this approach, attacks were detected with a 99.99% accuracy rate via binary classification, and an 87.06% accuracy rate via multi-class classification [23]. Evmorfos et al. (2020) examined streamlined methods for spotting SYN flood attacks in internet-connected devices. In particular, a DL stochastic neural network was executed on a virtual dataset named PCAP using 6 Wireshark for this prediction model, which was trained with normal traffic and a neural network with Long Short-Term Memory (LSTM). Ultimately, it was demonstrated that both false alarm rates and attack identifications were significantly enhanced by the proposed random neural network, which had an accuracy of 80.7% compared to 62.7% for the neural network with a large short-term memory [24]. Roopak et al. (2020) used an Intrusion Detection System (IDS) on the basis of the combination of a multi-objective optimization approach compatible with gene mutation to classify attacks from CNN-DL models with LSTM. As a result, this method achieved a Fl score of 99.36% and a high accuracy of 99.03% [25]. Dushimimana et al. (2020) developed a Bidirectional Recurrent Neural Network (BiRNN) for a high-durability security solution in IoT network security, which is an improvement of the application of the Recurrent Neural Network (RNN) and the gated RNN algorithms. The RF algorithm was also utilized in the 99KDDCup dataset to choose the relevant features of the model. At last, the BiRNN algorithm outperformed the RNN and gate RNN algorithms with an accuracy of 99.04% [26]. Malkawi et al. (2024) introduced a hybrid ML-driven IDS specifically designed for 5G Device-to-Device (D2D) communication in the IoT. The research combined RF, DT, and Support Vector Machine (SVM) classifiers to measure the detection process on a synthesized D2D communication dataset. Features were extracted and selected through a correlation-based method for better model efficiency and lower computational cost. The RF classifier achieved higher accuracy than the other models, proving its capability in identifying D2D-specific threats like impersonation and eavesdropping. The authors' framework was tested in a 5G network simulation, and the experiments indicated that the real-time usage of the system in the IoT infrastructure is quite feasible [27]. Almiani et al. (2021) employed the deep Kalman backpropagation neural network to identify DDoS attacks in 5G − IoT networks. The approach utilized the 2019 CICDDOS data set to carry out the implementation and experimentation of the model. Upon evaluating the model, it attained 94% detection accuracy in recognizing the attacks [28]. Setiadi et al. (2021) employed the Naive Bayes algorithm to detect DoS attacks in IoT devices; thus, a DoS attack on Raspberry Pi 3 was documented. The study reported that this algorithm, when applied to the data set, reached 64.02% accuracy in attack identification [29]. Churcher et al. (2021) used Artificial Neural Network (ANN), K-Nearest Neighbor (KNN), Logistic Regression (LR), SVM, DT, Simple Bayes, and RF algorithms for a detection system for Infiltration. When these algorithms were applied to the IoT-Bot dataset, the RF algorithm attained an accuracy of 99% in binary classification for spotting HTTP − DDoS attacks. For multi-class classification, the KNN algorithm outperformed the others, also reaching 99% accuracy [30]. Chesney et al. (2021) examined the accuracy of the logistic regression algorithm on the 2019 CICDDOS dataset to prevent IoT cybersecurity attacks. This method achieved 99.7% prediction accuracy in AD [31]. Singh Samom and Taggu (2021) used the RF algorithm to identify 4 types of DDoS attacks and utilized the 2019 CICDDOS and UNSW − 15NB data sets to compare the prediction results. As a result of this algorithm, it achieved 99.92667% detection accuracy in the 2019 CICDDOS dataset and 96.2% detection accuracy in the 15NB-UNSW dataset [32].

In most research, old data sets, such as 99KDDCUP, KDD-NSL, DARPA, 15NB-UNSW, etc., have been used for AD. Additionally, models trained on the previously mentioned old data sets tend to be less accurate. Therefore, the new CICIoT2023 dataset was utilized in this work to predict attacks on the IoT network. In this document, an optimized ML-based ID framework is aimed at being developed to enhance accuracy and responsiveness against diverse and evolving cyber threats. Prior studies have introduced various ID techniques in the IoT. While these methods usually target a particular type of threat, the network can still be susceptible to different attacks. The literature review showed that the proposed methods may have high overall accuracy, but the detection accuracy of each class in multi-class classification is low. Therefore, in this study, using optimized ML algorithms, an accurate model was presented to detect intrusive attacks on the IoT network.

Table 1 presents a comparative outline of recent SOTA approaches for ID in IoT environments, concentrating primarily on DDoS and SYN flood attacks. The table combines crucial data from specific studies, noting their chosen methods, experimental data collections, and stated assessment measures like precision and F1-score. As the table illustrates, DL approaches, namely, ResNet-based CNNs demonstrate high binary classification accuracy (up to 99.99%), yet their performance in multi-class scenarios—where precision drops to approximately 87%—raises concerns regarding their generalizability to diverse attack types. Similarly, conventional models like RF and KNN perform well in either binary or moderately multi-class settings but lack scalability to more granular attack classification.

Table 1: Comparative summary of SOTA-IoT-ID approaches

Reference	Method / Model	Dataset / Scenario	Accuracy	F1-score / Precision-Recall
Chen et al. (2020)	Graph-based GCN + correlation hybrid model	EuCNC 2020 IoT / real-world structures	–	**F1 up to 0.91**, with ≤ 2 % drop under 50 % loss (arXiv)
Hussain et al. (2020)	ResNet-based CNN (image-like traffic)	DoS/DDoS (11 attack types)	~99.99 % (binary)	~0.87 precision in multi-class (~11 classes)
Evmorfos et al. (2020)	Random Neural Network (RNN) vs LSTM	SYN-flood on IoT edge/fog	~96 % (RNN)	Low false alarm rate; LSTM much worse
Churcher et al. (2021)	KNN, RF, SVM, ANN, etc.	Bot-IoT dataset, binary vs multi-class	RF ≈ 99 % (binary), KNN ≈ 99 % (multi-class)	KNN ≈ 0.99 F1 (≈4 % above RF in multi-class)
Shakya & Abbas (2024)	XGBoost, SGD, KNN, Naive Bayes	DDoS detection in IoT networks	–	Report comparative F1/Precision/Recall, but no absolute numbers are given in the summary

This article is organized in the following manner: The opening section comprises the introduction. Section 2 details the research methods, along with a concise explanation of the primary ML classifiers and optimization techniques. Moreover, the dataset is outlined in section 3. The results of the study are presented in section 4, and the conclusion is available in section 5.

2 Methodology

This work aims to give a model to detect intrusive attacks on the IoT network using optimized ML models. The central research question addressed is:

- Can the proposed EXT-MFO hybrid model outperform existing ID models regarding accuracy and computational efficacy on the CICIoT2023 dataset?

According to this question, the primary hypothesis of the study is:

- Applying hybrid metaheuristic optimization algorithms to ensemble classifiers such as EXT will significantly improve detection accuracy and F1-score compared to conventional models.

Fig. 1 illustrates the research approach. Based on this illustration, the initial stage involves gathering data and preparing it. The data utilized in this research encompasses 33 cyber-attacks within an IoT structure comprising 105 items. Due to the extensive dataset and the high computational cost, 15,000 samples and 10 labels with the highest frequency were selected for this study. Data preprocessing is a crucial initial phase in creating an ML model to boost the quality and reliability of unprocessed data. This procedure encompasses refining and readying the data to make sure it is appropriate for the model, ultimately boosting its precision and effectiveness [33]. In this research, the subsequent preprocessing actions were utilized:

- Categorical to numerical conversion
- Feature correlation analysis to identify and remove highly correlated or irrelevant features
- Normalization through z-score standardization: omitting the mean and dividing by the standard deviation.
- Train-test split: arbitrarily dividing the data into a 25% testing group and a75% training group

When analyzing data collections, substantial disparities frequently exist between the highest and lowest values of features. Normalization is the organization of data that looks similar in all samples and features. There are various normalization techniques. To standardize the data in this research, the mean was deducted from each value, and the outcome was split by the standard deviation.

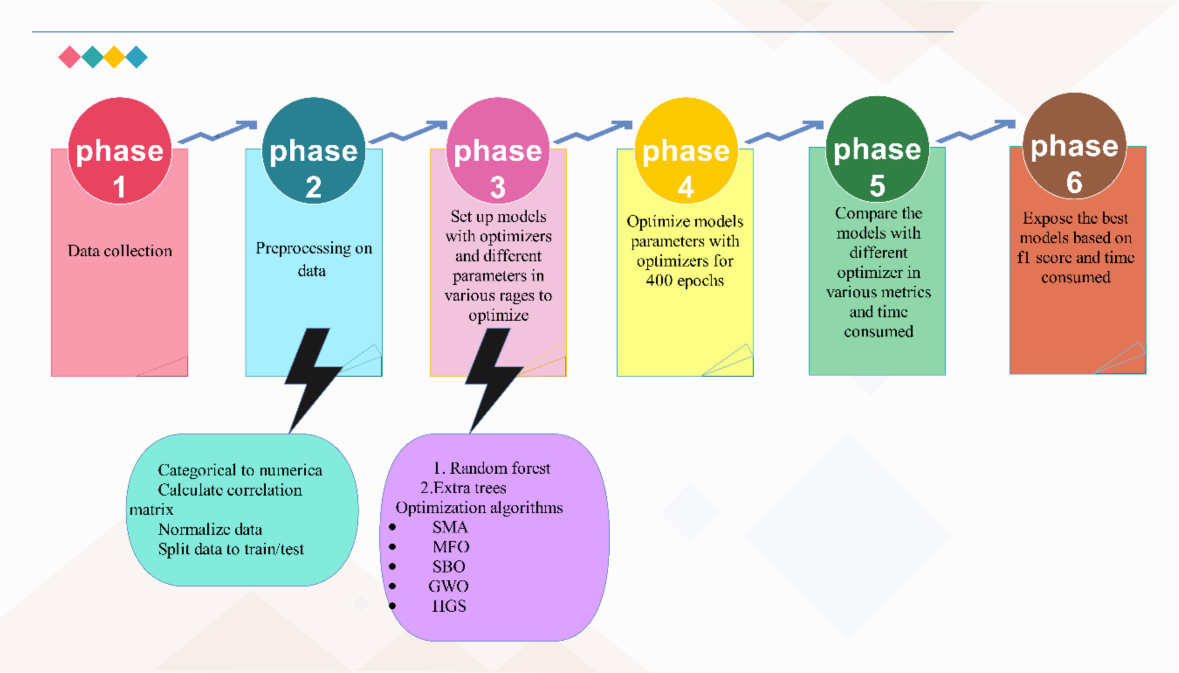

phase 1
Data collection
phase 2
Preprocessing on data
phase 3
Set up models with optimizers and different parameters in various rages to optimize
phase 4
Optimize models parameters with optimizers for 400 epochs
phase 5
Compare the models with different optimizer in various metrics and time consumed
phase 6
Expose the best models based on f1 score and time consumed
Categorical to numerica
Calculate correlation matrix
Normalize data
Split data to train/test
1. Random forest
2.Extra trees
Optimization algorithms
SMA
MFO
SBO
GWO
HGS

Figure 1: Flowchart of research methodology

In order to predict different classes based on selected features, two ML algorithms, including EXT and RF, were used as the main classifiers. Furthermore, to enhance the precision of forecasts, 6 optimization algorithms, including GWO, HGS, MFO, SBO, SMA, and WOA, were used to adjust the HPs of the main classifiers during 400 epochs. In other words, by combining different algorithms, 12 different models were presented for data classification. Finally, by using different evaluation indexes, incorporating Accuracy, Precision, F1 – score, and Recall, the precision of various models was put in contrast. In general, 4 different states evaluate the performance of different models. These states include True Negative (TN), FP, True Positive (TP), and False Negative (FN). Finally, based on these states, evaluation indices are determined from the following relations [34], [35], [36], [37], [38].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

$$F1\,Score = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{4}$$

2.1 Description of main classifiers

In the following, the main classifiers used in this study will be briefly described, including EXT and RF algorithms.

2.1.1 Extra trees (EXT)

Extremely Randomized Trees, also called the EXT algorithm, is an ensemble learning technique for regression and classification problems. It is based on the idea of a DT and, like RF and gradient boosting, is a member of the tree-based ensemble model family. A subset of characteristics called EXT is used for segmentation in RF depending on tree development. Additive trees select split sites for each feature at random, in contrast to RF, which takes into account a random selection of characteristics to divide at each node. The model's generalizability is improved, and possible over-fitting is reduced because of this random feature selection. EXT produces many candidates split points at random for each specified feature and then chooses the optimal split. In regression tasks, the ultimate prediction is determined by averaging the individual tree predictions, whereas in classification tasks, the individual tree predictions are combined using a majority vote [39], [40], [41], [42].

2.1.2 Random forest (RF)

Multiple DTs make up the categorization process known as RF. Bryman introduced the RF model in 2001 as a fresh approach to DT building. Every DT in this approach is constructed using a random vector. Using sample tree data, this algorithm makes predictions before acting on them. Every sample receives one vote from each tree, and a majority vote determines the outcome. The prognosis becomes more accurate the more trees there are in the forest [43], [44]. Data is loaded and preprocessed using RF to prepare the dataset. Subsets of the original dataset are then chosen at random (with replacement) to produce numerous training sets. This technique is referred to as bagging or bootstrap aggregation. A subset of characteristics is used to grow the DT for each bootstrap instance. A random feature subset is chosen at each node in the tree. Make a collection of DTS by iterating over the preceding step. It is possible to set the HP of the number of DT to grow. Together, the trees in this collection create an RF. Every tree in the RF makes an individual prediction about the target variable using the input characteristics. In classification assignments, the class with the highest number of votes from the decision trees establishes the final forecast. For regression problems, the average or middle value of the tree predictions is used as the final forecast [45], [46].

2.2 Description of optimizers

In the following, the optimizers used in this study will be briefly described, including GWO, HGS, MFO, SBO, SMA, and WOA algorithms.

2.2.1 Grey wolf optimizer (GWO)

The GWO algorithm, a meta-heuristic optimization method, was designed on the basis of wild grey wolves' social behaviors and hunting strategies. Mirjalili et al. introduced this concept in 2014. The GWO method is frequently employed in the resolution of optimization issues, wherein the target solution for a specific objective function is sought. It imitates how alpha, beta, and delta wolves work together and coordinate their hunting activities, much like grey wolves do [47]. The first step of the process is to randomly initialize a population of grey wolves, which stands in for possible fixes for the optimization issue. The grey wolves continuously adjust their locations based on their current spots and the locations of the alpha, beta, and delta wolves. The alpha wolf symbolizes the top solution achieved so far, while the beta wolf represents the second-best, and the delta wolf signifies the third-best. Throughout the quest, the grey wolves engage in both exploration and exploitation. Exploitation is accomplished by following the alpha, beta, and delta wolves into promising areas of the search space, whereas exploration is accomplished via random search and diversity preservation. In mathematical contexts, equations serve to modify the locations of grey wolves by replicating their predatory habits. The alpha, beta, and delta wolf locations are taken into consideration while adjusting the positions, along with a few other random exploration elements. The method keeps updating the grey wolf locations until a certain number of iterations pass or a termination requirement is satisfied. Numerous optimization issues, such as function optimization, feature selection, data clustering, and neural network training, have been addressed using the GWO method. It often produces favorable outcomes relative to other meta-heuristic methods and is well-known for its straightforwardness and robust convergence characteristics [48], [49].

2.2.2 Hunger games search (HGS)

The HGS method is a population-based approach that integrates aspects of rivalry and collaboration among agents to discover the best answers. The primary source of ideas for this method came from how animals work together in the wild to locate sustenance when they are in need of it. HGS relies on the activities and preferred behaviors of animals motivated by their need for food. By centering it on the basic principle that "Hunger" acts as the primary homeostatic drive and reason behind all animal behaviors, choices, and activities, this active, fitness-based search approach renders the optimization procedure more understandable and reliable for new users and those in charge. To replicate the impact of hunger on each search phase, The HGS incorporates the concept of hunger into the feature extraction procedure. To put it another way, an adjustable weight is created and used in accordance with the idea of hunger. It adheres to the computational logic activities that the majority of species participate in. These competitive events and games frequently serve as flexible evolutionary strategies intended to raise the likelihood of survival and foraging. This approach's dynamic nature, simple structure, fast convergence performance, and acceptable solution quality make it more efficient than the optimization strategies now in use [50], [51], [52].

2.2.3 Moth flame optimization (MFO)

The MFO is a meta-heuristic algorithm influenced by nature, developed by MirJalili in 2015. It draws its inspiration from the way moths navigate in relation to a flame. This algorithm is fundamentally based on the fact that Moths, at night, change the direction of their movement according to the moonlight, and thus they end up moving in a straight line [53]. The study of MFO remains a dynamic field because of its unique inspirations and good applications in the research and development community. The MFO algorithm has become well-known in the optimization field due to its distinctive sources of inspiration and strong performance. The current research is aimed at increasing the algorithm's capacity, finding new places where it can be applied, and also investigating the nature of this optimization method. The MFO algorithm proceeds as such [54], [55]:

- Initialization: The process commences by generating a haphazard group of moths (possible solutions) inside the defined area.
- Fitness Assessment: Each moth's performance is gauged by employing the objective function tied to the optimization challenge.
- Classification and Flame Choice: The moths are categorized according to their fitness scores, and the moth with the highest score is chosen to be the "flame," representing the optimal solution identified up to that point.
- Spiral Movement: Each moth updates its position by performing a spiral movement around the flame, mimicking the phototaxis behavior of moths.
- Adaptive Mechanism: The spiral movement is dynamically adjusted based on the distance between the moth and the flame, allowing for a balance between exploitation and exploration.
- Termination Condition: The algorithm proceeds to modify the locations of both the moths and the flame until a stopping point is reached. This could be based

on a limit to the number of cycles or achieving a particular degree of solution effectiveness.

2.2.4 Satin bowerbird optimization (SBO)

Samareh Moosavi and Khatibi Bardsiri developed the SBO in 2017, drawing inspiration from the reproductive and dwelling habits of the satin bowerbird (Ptilonorhynchus violaceus). The method has proven to perform competitively when measured against other meta-heuristic algorithms. Its main benefit is that it is easy to construct, requiring little tweaking of its few parameters. Essential features of the SBO [56], [57]:

1. Bower Building: The male satin bowerbird constructs elaborate bowers (nest-like structures) to attract and impress potential mates. The SBO algorithm simulates this bower-building behavior to optimize solutions.
2. Mate Selection: The female satin bowerbird selects the male with the most attractive and well-decorated bower as her mate. The SBO algorithm mimics this mate selection process to guide the search towards better solutions.
3. Dynamic Bower Modification: The male satin bowerbird continuously modifies and decorates his bower to make it more attractive. The SBO algorithm dynamically adjusts the solutions (bowers) during the optimization process.
4. Exploration and Exploitation: The algorithm finds a balance between exploration (looking for novel possible solutions) and exploitation (enhancing the current optimal solutions) through the use of changing connections between bowers and the mating procedure.

2.2.5 Slime mould algorithm (SMA)

Chen et al. had the idea of using the nature and behavior of a slime mold to create a new algorithm that is based on the population, and they named it the SMA in 2020. In that case, the slime mold utilized its senses in the form of the olfactory organ and the food odor emitted in the air to locate the prey. The algorithm is also recorded in the documentation along with the foraging behavior and the intelligent decision-making of the slime-blind, Physarum polycephalum. Main Features of the SMA [58], [59], [60]:

5. Foraging Behavior: The algorithm mimics the foraging behavior of the slime mold, which involves not only searching but also establishing an efficient network of interconnected tubes for converting the nutrients from the food sources(surface) to the organism.
6. Adaptive Network Formation: The slime mold is capable of changing its network of tubes to carry nutrients in the most efficient way, which is reflected in the algorithm's behavior.
7. Decision-Making: The slime mold demonstrates smart decision-making abilities that include solving the shortest route to the food that is represented in the algorithm.
8. Dynamic Adjustment: The algorithm dynamically changes the parameters and strategies in the course of the optimization process to ensure that it is exploring and exploiting in a balanced way

The SMA is a fresh entrant in nature-inspired optimization algorithms, but because of its innovative features and good performance, it still represents a vibrant research domain.

2.2.6 Whale optimization algorithm (WOA)

The WOA is a new nature-inspired metaheuristic optimization algorithm developed in 2016 by Seyedali Mirjalili and Andrew Lewis. The algorithm draws an analogy to the foraging behavior of the humpback whales, mainly the "bubble-net feeding" technique that they use. The main WOA characteristics are [61], [62], [63], [64]:

- Bubble-net Attacking Method: The algorithm is an imitation of humpback whales' bubble-net feeding behavior, in which the whales swim in a spiral and create a bubble wall to catch the fish.
- Exploration and Exploitation: The algorithm goes back and forth between exploration (looking for new solutions) and exploitation (improving the best solutions so far) stages, just like other swarm intelligence algorithms.
- Mathematical Model: The algorithm is designed with a series of mathematical expressions that represent the whale's foraging nature. These include spiral updating and shrinking encircling operations.

In general, the WOA represents a major contribution to the area of swarm intelligence and has become very popular among research community members due to its simple, efficient, and interesting application in practice.

All the optimizers were developed in Python 3.10 and run on a computer with an Intel Core i7-11800H CPU, 32 GB RAM, and Windows 11 OS. No GPU was involved. To make sure the comparison of time complexity was fair, all the classifiers and the optimizers were executed under memory constraints of ≤32 GB RAM and single-threaded CPU processing. A population size of 30, a maximum of one hundred iterations, and a convergence tolerance of 1e-6 were used for each algorithm. The runs were done with the same random seed to guarantee consistency. The optimization was carried out through wrapper-based tuning to pick the best HP for each classifier (EXT or RF).

Two metaheuristic nature-inspired optimization algorithms, MFO and SMA, were utilized to optimize EXT and RF classifiers, respectively, for hyperparameter tuning. During the tuning process, the following parameter ranges were investigated:

– Population size ∈ {30, 50, 100}
– Maximum iterations ∈ {100, 200, 300}
– For MFO: flame coefficient ∈ {0.5, 1.0}
– For SMA: control parameter a ∈ {1.5, 2.0, 2.5}

The final values were selected based on the best average F1-score obtained through 5-fold cross-validation. To ensure reproducibility, all experiments were executed with a fixed random seed of 42.

3 Description of the dataset

The CICIoT2023 dataset applied in this research comprises a collection of attacks directed at IoT devices. This dataset encompasses 33 distinct cyber-attacks within an IoT framework composed of 105 devices [65]. Data regarding various kinds of IoT infiltration contains multiple subcategories. The dataset encompasses 1,191,264 examples of network intrusion and 47 distinct attributes for each intrusion. This information can be employed to create a predictive model that distinguishes between different types of invasive attacks. Furthermore, the data can aid in the development of an IDS. To enhance reproducibility, all preprocessing and implementation steps are explicitly outlined. One-hot encoding was utilized to transform categorical attributes, while z-score standardization was applied to normalize numerical attributes. The preprocessing and model development processes were implemented in Python 3.10 using the Scikit-learn (v1.2.2), NumPy (v1.23.0), and Pandas (v1.5.3) libraries.

To reduce computational cost while maintaining diversity, 15,000 samples were picked from the original dataset through stratified sampling. The 10 classes with the highest occurrence frequency, including both benign and attack types, were chosen. This approach ensures class balance in the reduced dataset and avoids sampling bias while preserving key statistical characteristics of the original data distribution.

Fig. 2 illustrates the IoT arrangement employed to develop the CICIoT2023 dataset, which is made up of 105 IoT devices. There were 38 Zigbee and Z-Wave devices linked to 5 hubs, with 67 IoT devices directly involved in the attacks. This setup replicates how IoT products and services could actually be set up within a smart home environment. The detailed gadgets include a range of smart cameras, sensors, home devices, and micro-controllers, all linked and designed to monitor attack behaviors and enable the performance of repeated attacks. The lab includes different hardware and software, which allows for the execution of numerous attacks and the gathering of both harmless and harmful data.

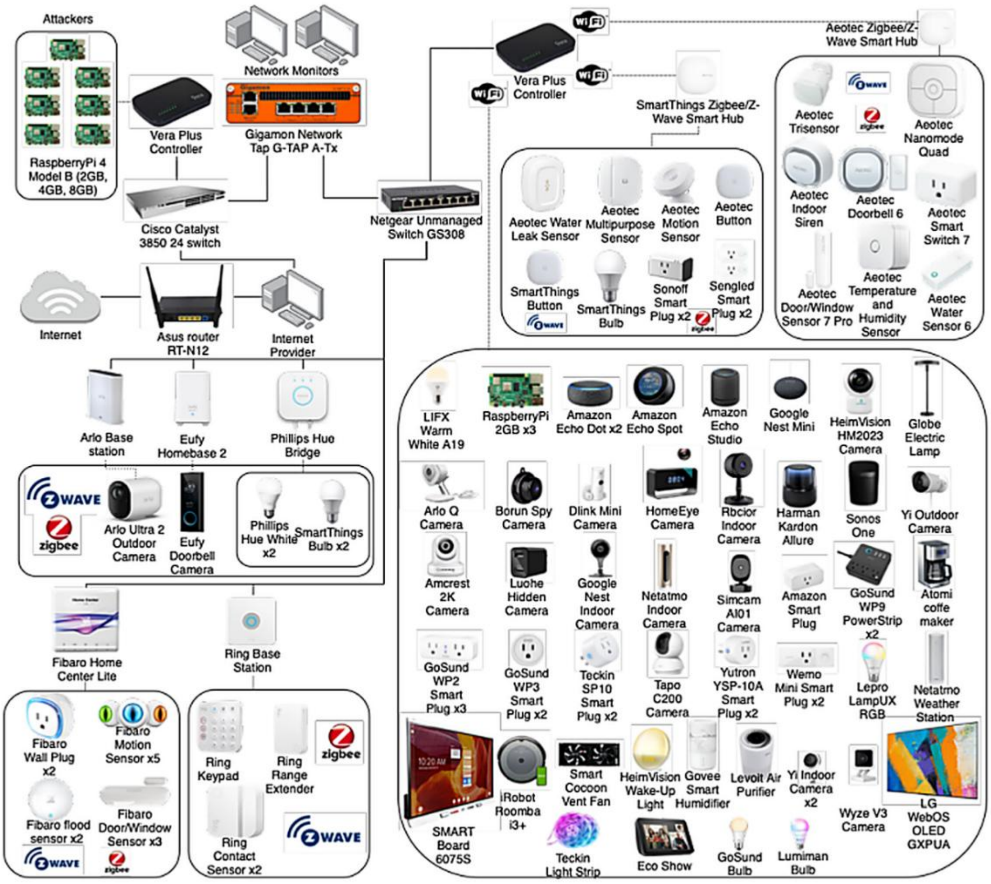


Figure 2: The IoT network topology employed in the experimental procedures

Threats that overwhelm an IoT system's availability are referred to as cyber-attacks. When DoS assaults occur, it is one Raspberry Pi that causes an overload of IoT devices. Furthermore, using an SSH-based master-client setup, many Raspberry Pis are utilized to carry out DDoS attacks.

One massive DDoS that can target IoT devices is the Mirai assault. Fig. 3 illustrates the use of 5 different Raspberry Pi devices to conduct multiple iterations of Mirai attacks. These elements are linked to the associations examined throughout the various tiers of the IoT system. A gateway enables an Internet connection, employing a Windows 10 system to oversee and provide web access. The attackers' connection to several IoT devices is made possible via an unmanaged Netgear switch. A variety of tools and a personalized Mirai configuration are used to carry out attacks. Smart speakers, cameras, sensors, and other IoT devices are all under the supervision of an online supervisor. Remarkably, Mirai has not been a part of the attack sets of some earlier research. This study centers on various potential hazards to IoT devices and explores the examination and enactment of innovative IoT attacks, including those that utilize emerging protocols.

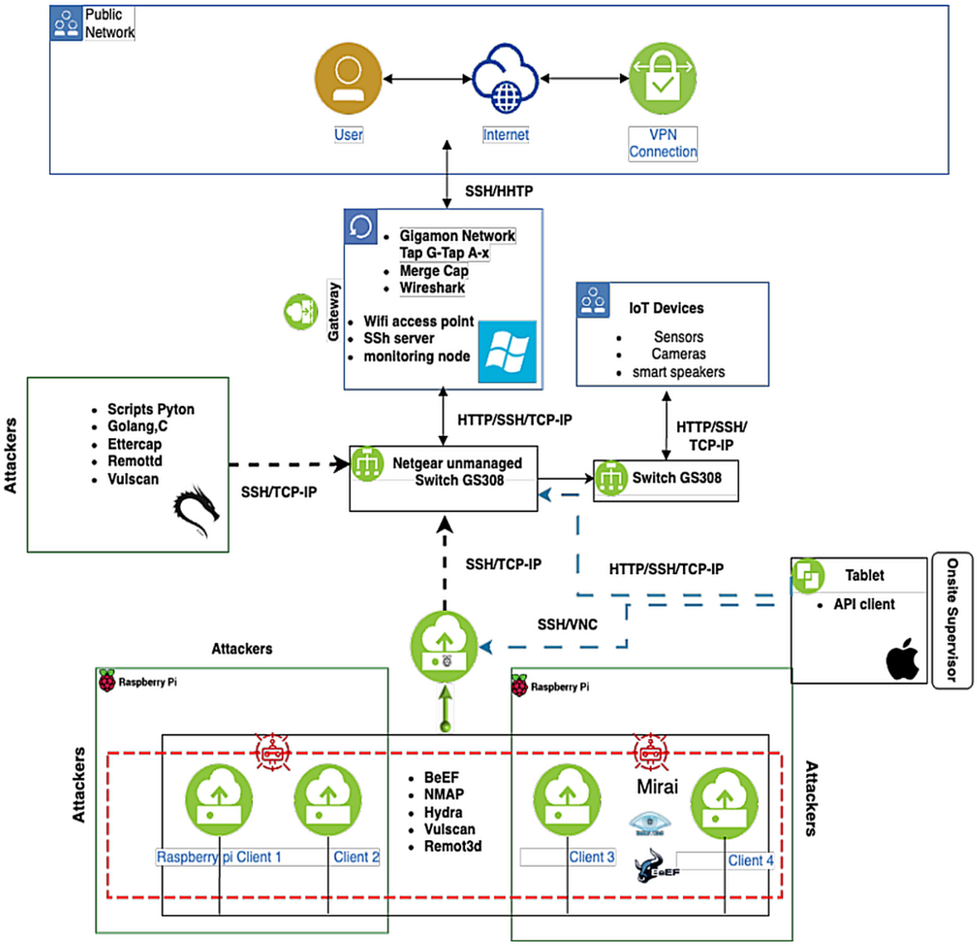


Figure 3: Fundamental attack structure for the dataset

Table 2 presents a comprehensive catalog of each characteristic integrated into the dataset. It highlights the qualities related to these features. The table delineates the mean, standard deviation (Std), minimum (min), 25th percentile (25%), median (50%), 75th percentile (75%), and maximum (max) values for each feature throughout the entire dataset.

Table 2: Dataset overview

Feature	Mean	Std	Min	25%	50%	75%	Max
flow_duration	5.76544939	285.034171	0	0	0	0.10513809	394,357.207
Header_Length	76,705.9637	461,331.747	0	54	54	280.555	9,907,147.75
Protocol Type	9.06568989	8.94553292	0	6	6	14.33	47
Duration	66.3507169	14.0191881	0	64	64	64	255
Rate	9064.05724	99,562.4906	0	2.09185589	15.7542308	117.384754	8,388,608
Srate	9064.05724	99,562.4906	0	2.09185589	15.7542308	117.384754	8,388,608
Drate	5.46 × 10−6	0.00725077	0	0	0	0	29.7152249
fin_flag_number	0.08657207	0.28120696	0	0	0	0	1

syn_flag_number	0.20733528	0.40539779	0	0	0	0	1
rst_flag_number	0.09050473	0.28690351	0	0	0	0	1
psh_flag_number	0.08775006	0.28293106	0	0	0	0	1
ack_flag_number	0.12343168	0.32893207	0	0	0	0	1
ece_flag_number	1.48 × 10−6	0.00121571	0	0	0	0	1
cwr_flag_number	7.28 × 10−7	0.00085338	0	0	0	0	1
ack_count	0.09054283	0.28643144	0	0	0	0	7.7
syn_count	0.33035785	0.6635354	0	0	0	0.06	12.87
fin_count	0.09907672	0.32711642	0	0	0	0	248.32
urg_count	6.23982356	71.8524536	0	0	0	0	4401.7
rst_count	38.4681213	325.384658	0	0	0	0.01	9613
HTTP	0.04823423	0.21426079	0	0	0	0	1
HTTPS	0.05509922	0.22817383	0	0	0	0	1
DNS	0.00013068	0.01143079	0	0	0	0	1
Telnet	2.14 × 10−8	0.00014635	0	0	0	0	1
SMTP	6.43 × 10−8	0.00025349	0	0	0	0	1
SSH	4.09 × 10−5	0.00639772	0	0	0	0	1
IRC	1.50 × 10−7	0.00038722	0	0	0	0	1
TCP	0.57383427	0.49451846	0	0	1	1	1
UDP	0.21191758	0.40866676	0	0	0	0	1
DHCP	1.71 × 10−6	0.00130903	0	0	0	0	1
ARP	6.62 × 10−5	0.00813521	0	0	0	0	1
ICMP	0.16372157	0.37002273	0	0	0	0	1
IPv	0.99988731	0.01061485	0	1	1	1	1
LLC	0.99988731	0.01061485	0	1	1	1	1
Tot sum	1308.32257	2613.30273	42	525	567	567.54	127,335.8
Min	91.6073456	139.695326	42	50	54	54	13,583
Max	181.963418	524.030902	42	50	54	55.26	49,014
AVG	124.668815	240.991485	42	50	54	54.0497296	13,583
Std	33.3248065	160.335722	0	0	0	0.37190955	12,385.2391
Tot size	124.691567	241.549341	42	50	54	54.06	135,83
IAT	83,182,525.9	17,047,351.7	0	83,071,566	83,124,522.4	83,343,908	167,639,436
Number	9.49848933	0.81915318	1	9.5	9.5	9.5	15
Magnitue	13.12182	8.62857895	9.16515139	10	10.3923048	10.3967148	164.821115
Radius	47.0949848	226.769647	0	0	0	0.50592128	17,551.2708
Covariance	30,724.3565	323,710.68	0	0	0	1.34421569	154,902,159
Variance	0.0964376	0.23300	0	0	0	0.08	1
Weight	141.51237	21.0683073	1	141.55	141.55	141.55	244.6

Table 3 compares the CICIoT2023 dataset with a number of different datasets. The results of this table show the superiority of CICIoT2023 over others. Therefore, this dataset was used in this study.

Table 3: CICIoT2023 contributions compared to current IoT security datasets

	Extensive Topology (> 100 *Devices*)	*Execution of* 33 *Attacks Divided into* 7 *Classes*	ML and DL Evaluation
IoTHIDS			
N − BaIoT			✓
Kitsune			✓
IoTNIDS			✓
IoT − SH			✓
BoT − IoT			✓
MedBIoT			✓
IoT − 23 (2020)			✓
IoTIDS			✓
MQTT			✓
MQTT − IoT − IDS			✓
X − IIoTID			✓
WUSTL − IIoT			✓
Edge − IIoTSet			✓
CICIoT2023	✓	✓	✓

The main dataset contains 34 different labels. In this notification, due to the large amount of data and the cost of calculations, 15,000 samples and labels were used with the highest frequency. In Fig. 4, the ten labels with the highest frequency used in this study are shown separately from the test and training data. These ten labels represent the types of intrusive attacks on IoT devices. According to this figure, DDoS − ICMP_Flood, DDoS − UDP_Flood, and DDoS − CP_Flood classes have been the most frequent among different types of attacks, respectively.

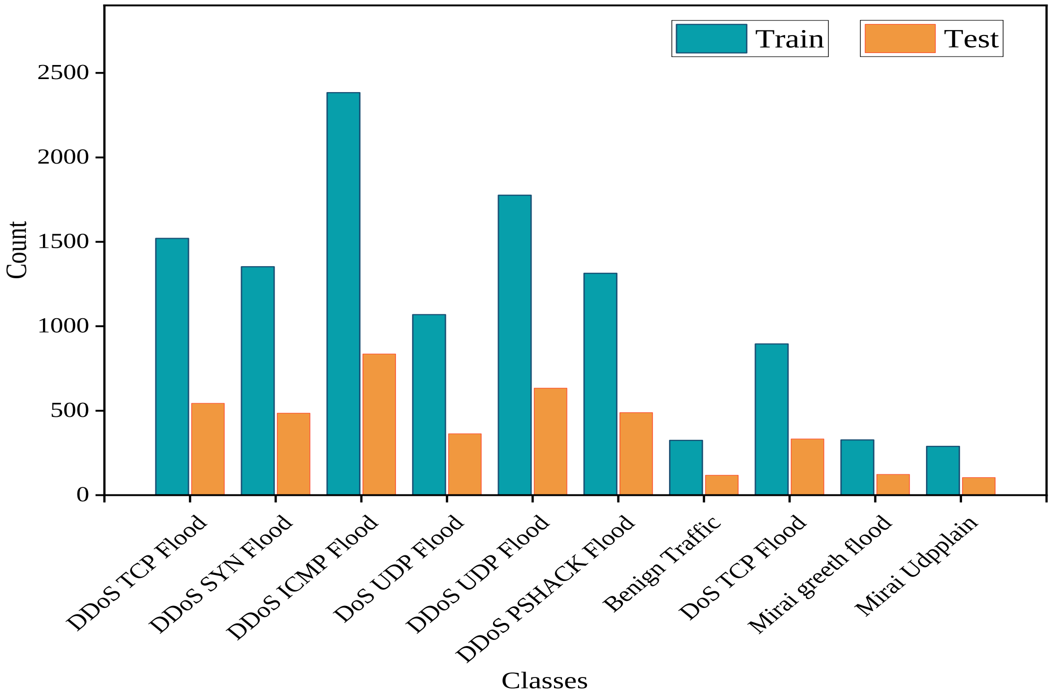


Figure 4: Bar plot related to the number of classes

In Fig. 5, the correlation between features and different labels is shown. The higher the value of the correlation between features and labels, the greater the relationship between them. According to this figure, it can be seen that Magnitude, AVG, and Tot sum features have the most positive correlation with different labels. Also, Rate, LLC, and Weight features have the highest negative correlation with different labels.

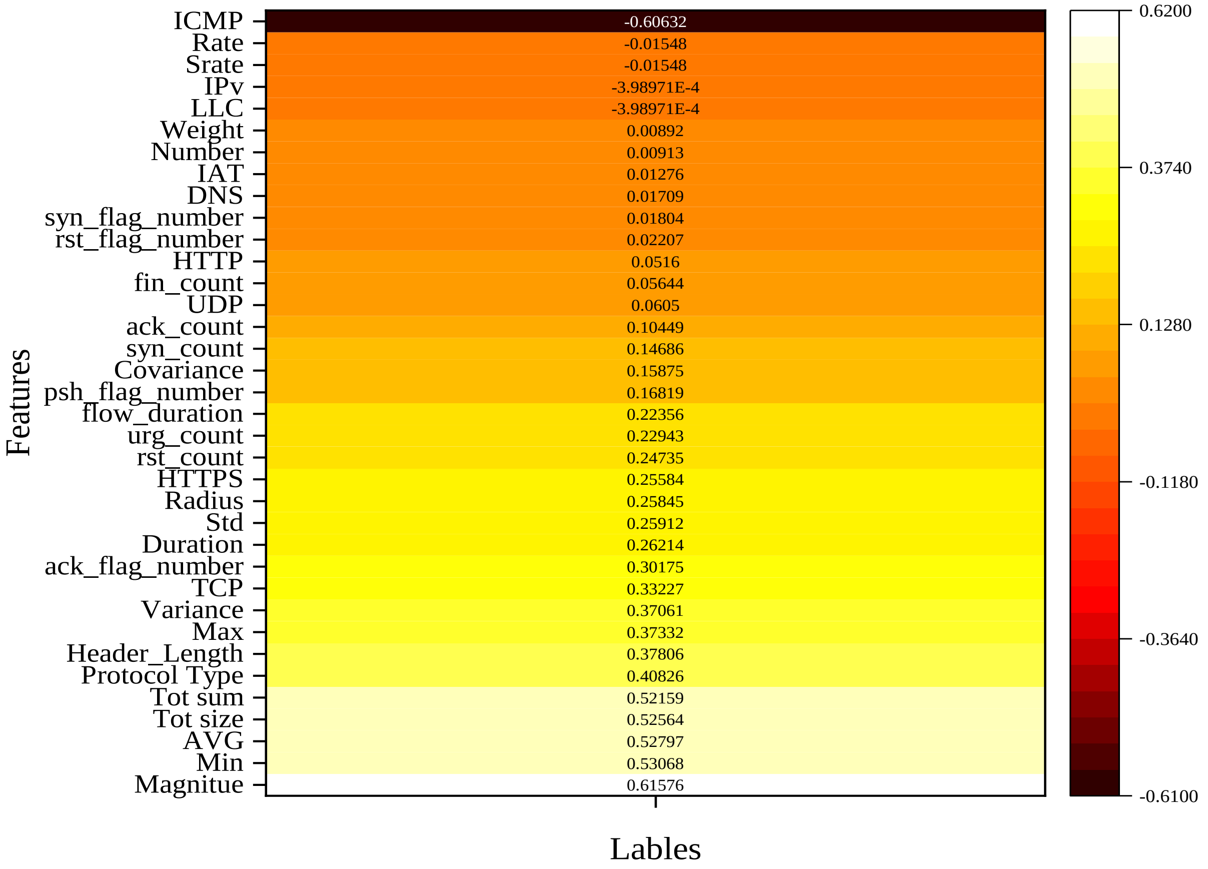


Figure 5: Correlation between different features and labels

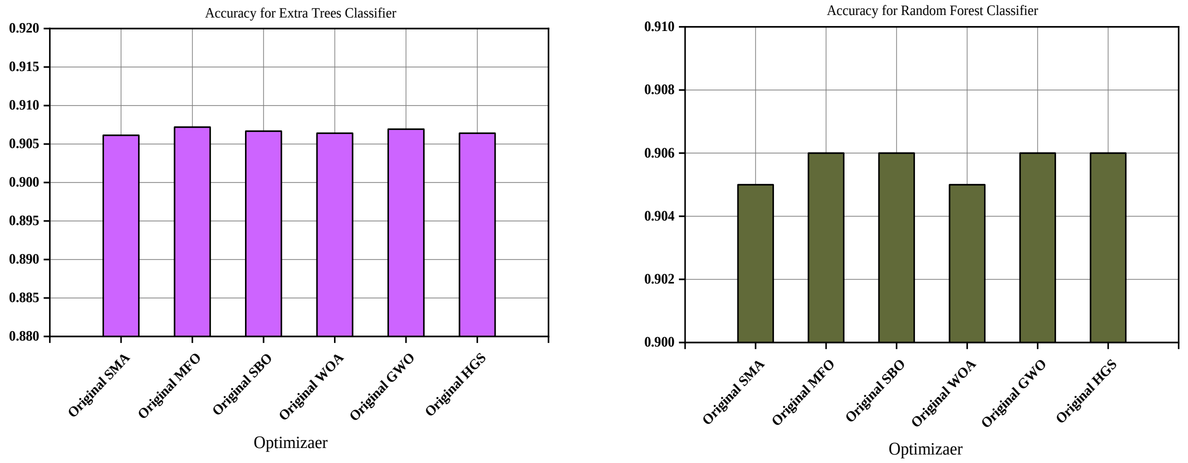
4 Results

In this section, the accuracy of different approaches is analyzed based on evaluation indices. Table 4 shows the values of different evaluation indices separately for each of the models. In contrast, for the RF classifier, the MFO, SBO, GWO, and HGS optimizers achieved the highest accuracy scores.

Table 4: Evaluation indices related to all models

Models	Accuracy	Precision	Recall	F1
EXT-GWO	0.906933	0.929214	0.899884	0.90505
EXT-HGS	0.9064	0.928572	0.899545	0.904632
EXT-MFO	0.9072	0.929827	0.900082	0.905283
EXT-SBO	0.90666	0.928892	0.899715	0.904841
EXTSMA	0.9061	0.92886	0.8988	0.903776
EXT-WOA	0.9064	0.930965	0.897985	0.903779
RF-GWO	0.906	0.93	0.896	0.901
RF-HGS	0.906	0.93	0.896	0.901
RF-MFO	0.906	0.93	0.896	0.901
RF-SBO	0.906	0.93	0.896	0.901
RF-SMA	0.905	0.929	0.896	0.901
RF-WOA	0.905	0.927	0.896	0.9

Given the high cost of FPs and FNs in the nature of ID, the F1 − score is emphasized in the assessment as it balances precision and recall, 2 metrics crucial for imbalanced or critical classification tasks. In order to compare more easily, Fig. 6 demonstrates the process comparison of EXT and RF classifiers using the Accuracy metric. According to this figure, based on the EXT classifier, the MFO optimizer has the highest value in the accuracy index. In other words, among different optimizers, the MFO is the most accurate in optimizing and adjusting the HPs of the EXT classifier.

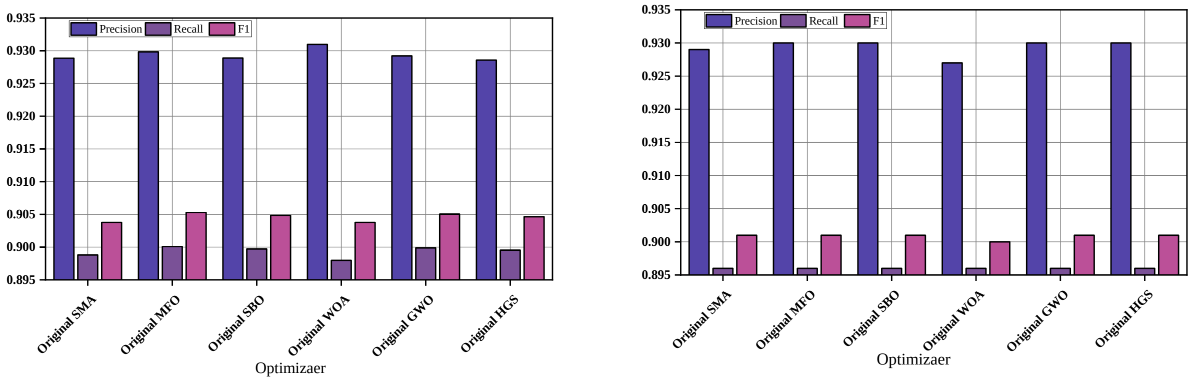


(a) EXT Classifier

(b) RF Classifier

Figure 6: Performance of all models, assessed utilizing the Accuracy index

The Recall, Precision, and F1-score index values for the different classifiers are also displayed in Fig. 7. Based on the EXT classifier, as demonestrated in Table 4 and demonstrated in Fig. 7, the WOA optimizer achieves the highest accuracy among the EXT-based models, confirming its effectiveness in parameter tuning. Nonetheless, the MFO optimizer has outperformed others according to the F1 – Score and Recall indexes. The figures for the Recall and F1 – score indices associated with EXT-MFO are 0.9052 and 0.9, respectively, and they are all greater than the comparable figures in other models. In contrast, the MFO, SBO, GWO, and HGS optimizers have the greatest Recall, Precision, and F1 – score index values when it comes to RF classifiers. Therefore, these optimizers are more accurate in optimizing the HPs of RF.



(a) EXT Classifier

(b) RF Classifier

Figure 7: Performance of all models assessed utilizing Precision, Recall, and F1 – score metrics

Fig. 8 presents the F1-score results for the evaluated models, enabling a comparison of classification effectiveness. According to this figure, it can be seen that, in general, the EXT classifier, in combination with different optimizers, is more accurate than RF. Among the different modes of the EXT-MFO hybrid model, it has more F1-score values than others. Therefore, this model has the most accuracy in identifying internal attacks on the IoT network.

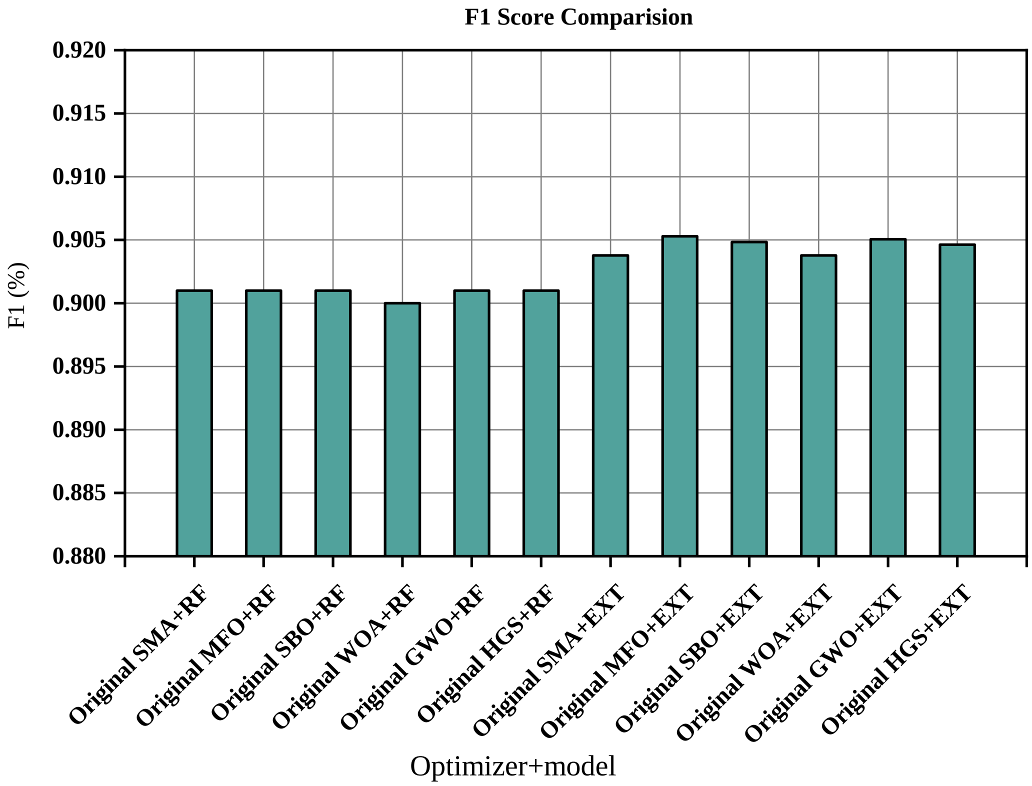


Figure 8: Evaluating models' accuracy using the F1-score index

The precision of various models is contrasted according to time spent in Fig. 9. This chart shows that models built on EXT classifiers combined with various optimizers generally spend more time tuning HPs. According to the outcomes illustrated in Fig. 9, the WOA optimizer consumed more time for hyperparameter tuning when applied to the EXT classifier compared to the RF classifier, indicating a higher computational cost for EXT-based models under this optimizer. Also, the maximum and minimum time spent on optimizing HPs are related to EXT-HGS and RF-SMA hybrid models, respectively.

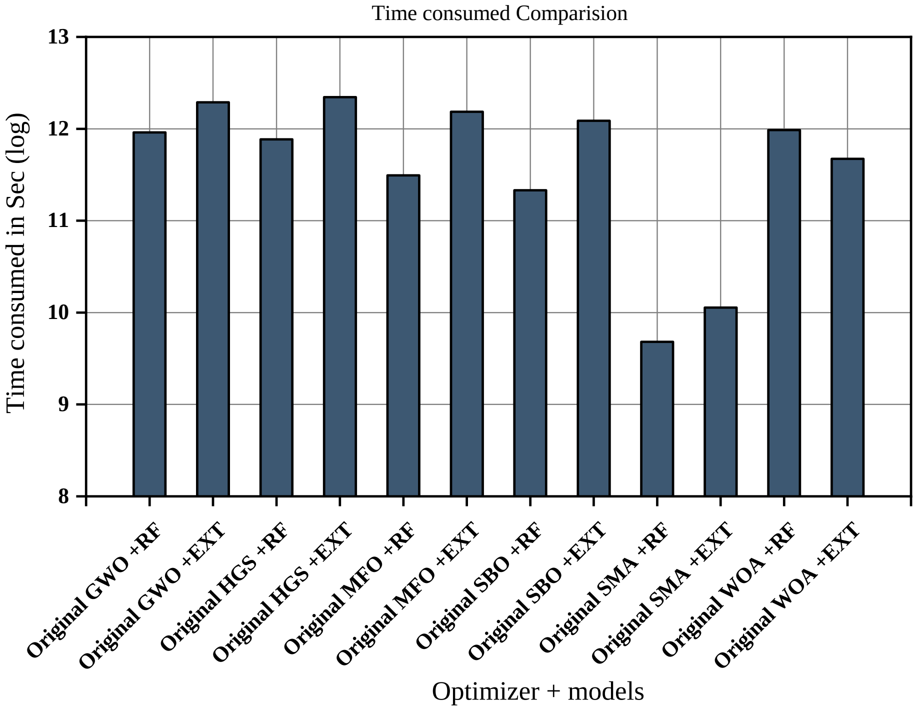


Figure 9: Comparing the accuracy of models based on the time consumed.

Figs. 10 and 11 show the confusion matrices of the best optimizer for EXT and RF classifiers, i.e., EXT-MFO and RF-SMA, respectively. In these figures, the horizontal axis shows the classes predicted by the model, and the vertical axis shows the observed classes. Fig. 10 clearly shows that the EXT-MFO model identified the DoS − TCP − Flood, DoS − UDP − Flood, DDoS − TCP − Flood, and DDoS − TCP − Flood classes with the largest inaccuracy. The number of wrong samples forecasted by the EXT-MF model for the mentioned classes is 162, 118, 12, and 52, respectively. However, according to Fig. 11, the number of wrongly forecasted samples by the RF-SMA model for the mentioned classes is 163, 142, 10, and 38, respectively.

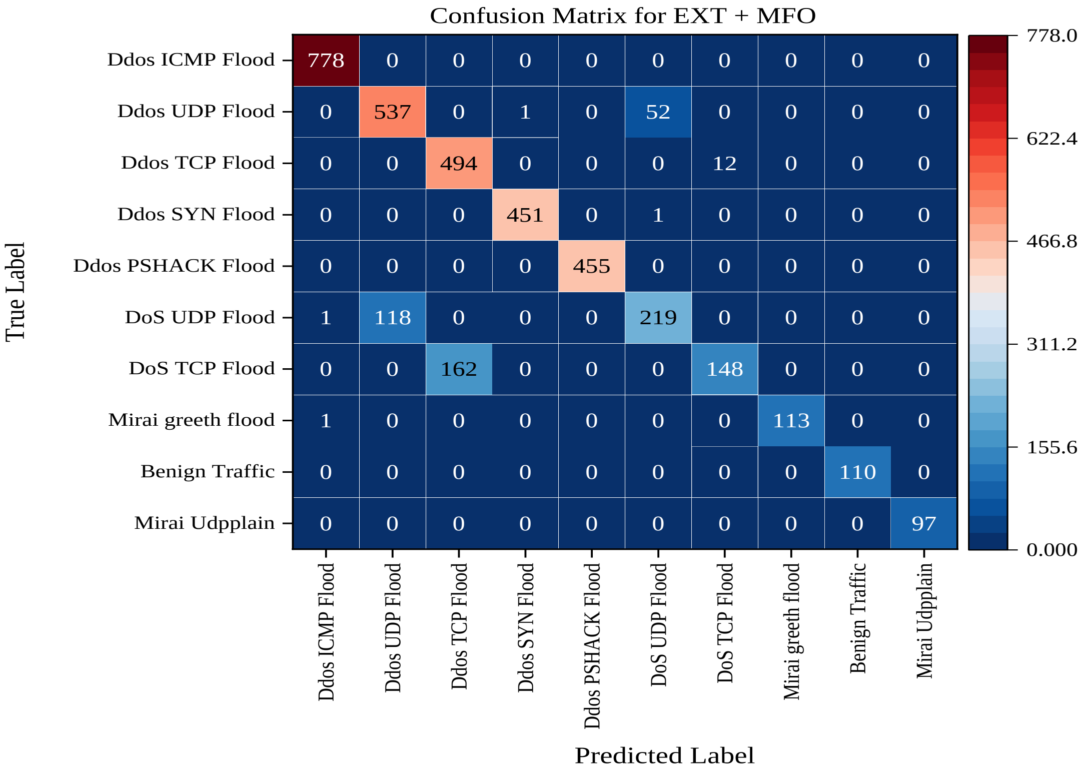


Figure 10: The confusion matrix related to the EXT-MFO model

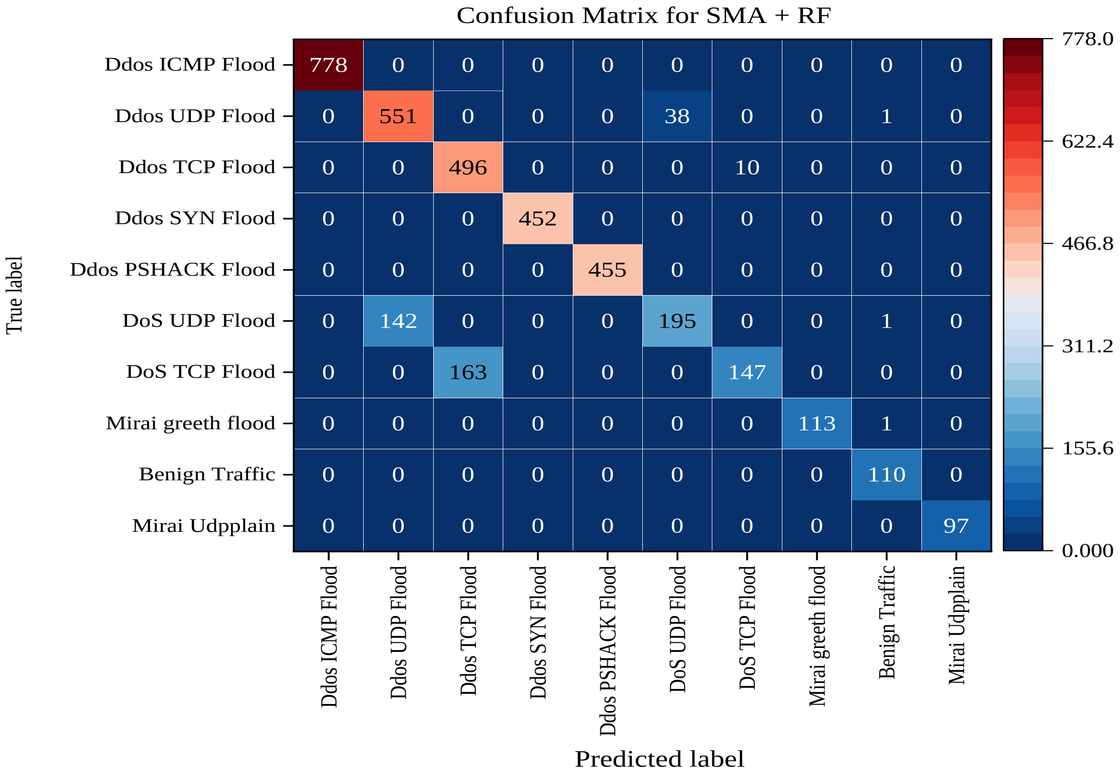


Figure 11: The confusion matrix related to the RF-SMA model

The confusion matrices (Figs. 10 and 11) reveal that the most frequent misclassifications occur among flooding-based attacks, particularly DoS-TCP-Flood, DoS-UDP-Flood, and DDoS variants. This indicates that the models struggle to differentiate between these categories because of similar statistical and traffic-related characteristics. Despite this, the EXT-MFO model exhibits slightly fewer misclassifications than RF-SMA, indicating stronger generalization. These observations highlight the challenge of accurately differentiating between semantically similar attack types and reinforce the importance of incorporating more discriminative features in future work.

These patterns suggest that overlapping traffic patterns or similar feature distributions among DoS and DDoS attack types may lead to difficulty in differentiation. In particular, the DoS-TCP-Flood and DoS-UDP-Flood classes often share packet frequency and duration characteristics, which might confuse the classifiers. From an optimization perspective, MFO appears to offer more stable convergence and better feature subset selection for the EXT classifier, potentially due to its spiral search mechanism and better exploration-exploitation balance. On the other hand, SMA's adaptive conduct in swarm motion could be more compatible with the structure of the RF model, which is a model that fundamentally gains from randomness and feature subspace sampling. These findings corroborate the importance of employing hybrid methods that are adjusted to the structure of the classifier and the complexity of the dataset, especially in cases with similar class labels.

5 Discussion

This section describes the process of the newly improved EXT-MFO hybrid model in relation to SOTA methods that were cited in the Related Works section. The EXT-MFO model reached 90.72% accuracy and an F1-score of 0.905; thus, it outperformed the other hybrid configurations and the literature benchmarks. The MFO algorithm had a great impact on this success since it efficiently updated the HP of the EXT classifier by choosing the best parameters. In addition, the robustness of the EXT model in handling high-dimensional data dealt well with the complicated patterns that were present in the CICIoT2023 dataset. When matched with most studies that employed traditional ML models such as XGBoost or SVM on datasets like NSL-KDD or TON_IoT, the proposed model yielded competitive or better results. Many of the methods that were developed, for example, the ones used in IoT attack classification tasks, reported F1-scores below 0.90. Especially when they were dealing with a broader variety of attack types and a high dimensionality of features, this was found to be the case. This makes a good point for the proposed hybrid model as it illustrates its ability to solve these problems both through the structural design and optimization strategy.

It has been recently observed that computational trade-offs exist between the classifiers used. In the case when both EXT and RF classifiers were applied, the models yielded high predictive performances, but the EXT-predicted models still generally needed more training time. Specifically, the time taken to train varied from 31 minutes to 40.5 minutes depending on the optimization algorithm used. While the RF-based models were equivalent to EXT ones in predictive performance, they were a bit faster but gave slightly lower F1-scores, which indicated a trade-off between accuracy and computational efficiency. This trade-off may still be present and influence some factors of the model choice, such as the application context. For instance, in real-time ID scenarios where a faster response is crucial, choosing the RF-based models will be advantageous. On the other hand, if offline analysis or situations where the highest detection accuracy is the main focus are considered, then the EXT-MFO setting fits best.

The utilization of the CICIoT2023 dataset marks a significant step forward in comparison with previous datasets, like NSL-KDD or Bot-IoT, due to its extensive and current attack profiles. It covers a broad range of latest IoT-specific threats (e.g., DDoS, information theft, backdoor exploits) and also records both network-level and application-level activities by means of a large feature set. This abundance allows for more accurate modeling of IoT traffic, as well as helps models become more flexible when they meet new types of attacks. Besides, the results on classifier performance with this dataset provide more practical insights into the real-world IoT security implementations. For example, the consistency in performance across multiple metrics suggests the feasibility of deploying lightweight ML-based ID in edge environments without compromising detection accuracy.

Overall, the findings emphasize that leveraging hybrid optimization in conjunction with ensemble learning can significantly enhance ID performance in complex IoT environments, while careful consideration must be given to computational constraints in deployment scenarios.

6 Conclusion

Over the past ten years, assaults targeting the IoT have emerged as a key subject field of cybersecurity as one of the security challenges of digital societies. With the expansion of the role of connected objects in daily life and the rising number of gadgets linked to the web, various attacks have also taken place to take advantage of the security weaknesses of these objects. In all information processing systems, the detection of cyber-attacks is considered a major challenge, and by timely detection of attacks, their effects can be blocked or reduced. The IoT system is not exempt from this phenomenon, and with the growing progress of this technology and the expansion of its infrastructure, the need for an intelligent IDS with high accuracy and speed is very important. In this study, an attempt was made to provide a model to identify intrusive attacks on the IoT network using optimized ML models. This research used a dataset comprising 33 cyberattacks within an IoT configuration that contained 105 units. Due to the large amount of data and the cost of calculations, 15,000 samples and 10 labels were used, with the highest frequency in this information. For this purpose, 2 ML algorithms, including EXT and RF, were used as the main classifiers. Additionally, to enhance the correctness of forecasts, 6 optimization algorithms, including GWO, HGS, MFO, SBO, SMA, and WOA, were used to adjust the HPs of the main epoch's classifiers. Finally, by using different evaluation indexes, including *Precision*, *Recall*, *Accuracy*, and *F1-score*, the accuracy of various models was compared. When comparing the accuracy of multiple models using the F1-score index, it was generally found that the EXT classifier, when used in conjunction with various optimizers, outperforms the RF classifier in terms of accuracy. Among the different modes, the EXT-MFO hybrid model has more F1-score values than the others. An evaluation of various performance metrics indicates that MFO and SMA are the most effective optimizers for the EXT and RF classifiers, respectively. Also, comparing the accuracy of numerous models based on the time spent to optimize HPs showed that, in general, models based on the EXT classifier in combination with different optimizers spend more time optimizing HPs. Therefore, the EXT-MFO hybrid model is proposed to detect intrusive attacks on the IoT network. Therefore, based on the experimental outcomes, the combination of the EXT classifier with the MFO optimizer, referred to as the EXT-MFO configuration, demonstrated slightly higher performance regarding F1-score and accuracy. However, the RF classifier, particularly when optimized with the SMA algorithm, achieved comparable results. These findings suggest that both classifiers can be effectively enhanced through suitable optimization strategies, and the choice may depend on specific application constraints such as computational cost or convergence behavior. Future work will explore model resilience under adversarial scenarios and dynamically evolving attack patterns, to ensure robust deployment in real-world IoT environments.

Nomenclature

Abbreviation	Explanation	Abbreviation	Explanation
ANN	Artificial Neural Network	LR	Logistic Regression
DDoS	Distributed Denial of Service	MFO	Moth Flame Optimization
DL	Deep Learning	MitM	Man-in-the-middle attack
DoS	Denial of Service	ML	Machine Learning
EXT	Extra trees	Res Net	Residual Network
FN	False Negative	RF	Random Forest
FP	False Positive	SBO	Satin Bowerbird Optimization
GWO	Grey Wolf Optimizer	SMA	Slime Mould Algorithm
HGS	Hunger Games Search	TN	True Negative
IoT	Internet of Things	TP	True Positive
KNN	K-Nearest Neighbor	WOA	Whale Optimization Algorithm

Acknowledgements

2022 Wuxi Vocational College of Science and Technology School level Teaching Innovation Team; Project Name: Software Technology Professional Teaching Innovation Team

Competing interests

No competing interests are disclosed by the authors.

Authorship contribution statement

Xiaonan Chen: Supervision, Conceptualization, Project administration, Writing-Original draft preparation.

Conflicts of interest

The writers assert that they have no competing interests concerning the release of this document.

Author statement

All authors have reviewed and endorsed the manuscript, confirming that it adheres to the authorship criteria outlined previously in this document, and each one is confident that the manuscript reflects genuine scholarly effort.

Ethical approval

The institutional review board has given its ethical endorsement to the research paper, which guarantees that participant rights are protected and that all relevant ethical standards are met.

References

[1] R. A. R. A. Mouha, "Internet of things (IoT)," *Journal of Data Analysis and Information Processing*, vol. 9, no. 02, p. 77, 2021.

[2] L. Khalid and L. Khalid, "Internet of Things (IoT)," *Software Architecture for Business*, pp. 107–127, 2020.

[3] F. A. Balouch, K. M. Wafa, and A. Ahmad, "Internet of things (iot), its application area and combination with gps," *Galaxy International Interdisciplinary Research Journal*, vol. 10, no. 1, pp. 725–734, 2022.

[4] M. Lombardi, F. Pascale, and D. Santaniello, "Internet of things: A general overview between

architectures, protocols and applications," *Information*, vol. 12, no. 2, p. 87, 2021.

[5] Y. Shah and S. Sengupta, "A survey on Classification of Cyber-attacks on IoT and IIoT devices," in *2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, IEEE, 2020, pp. 406–413. https://doi.org/10.1109/UEMCON51285.2020.9298138

[6] Y. Bin Zikria, R. Ali, M. K. Afzal, and S. W. Kim, "Next-generation internet of things (iot): Opportunities, challenges, and solutions," *Sensors*, vol. 21, no. 4, p. 1174, 2021.

[7] A. A. Laghari, K. Wu, R. A. Laghari, M. Ali, and A. A. Khan, "A review and state of art of Internet of Things (IoT)," *Archives of Computational Methods in Engineering*, pp. 1–19, 2021. https://doi.org/10.1007/s11831-021-09622-6

[8] I. Lee, "Internet of Things (IoT) cybersecurity: Literature review and IoT cyber risk management," *Future Internet*, vol. 12, no. 9, p. 157, 2020.

[9] A. E. Omolara, A. Alabdulatif, O. I. Abiodun, M. Alawida, A. Alabdulatif, and H. Arshad, "The internet of things security: A survey encompassing unexplored areas and new insights," *Comput Secur*, vol. 112, p. 102494, 2022. https://doi.org/10.1016/j.cose.2021.102494

[10] O. I. Abiodun, E. O. Abiodun, M. Alawida, R. S. Alkhawaldeh, and H. Arshad, "A review on the security of the internet of things: Challenges and solutions," *Wirel Pers Commun*, vol. 119, pp. 2603–2637, 2021. https://doi.org/10.1007/s11277-021-08348-9

[11] O. Kayode, "'Machine Learning Approaches to Improve Security and Performance Monitoring of IoT Devices,' ," *The University of Texas at San Antonio*, 2020.

[12] P. Shukla, C. R. Krishna, and N. V. Patil, "Iot traffic-based DDoS attacks detection mechanisms: A comprehensive review," *J Supercomput*, vol. 80, no. 7, pp. 9986–10043, 2024. DOI: 10.1007/s11227-023-05843-7

[13] Y. K. Saheed, A. I. Abiodun, S. Misra, M. K. Holone, and R. Colomo-Palacios, "A machine learning-based intrusion detection for detecting internet of things network attacks," *Alexandria Engineering Journal*, vol. 61, no. 12, pp. 9395–9409, 2022. https://doi.org/10.1016/j.aej.2022.02.063

[14] O. Vermesan *et al.*, "Internet of things strategic research and innovation agenda," in *Internet of things*, River Publishers, 2022, pp. 7–151.

[15] S. Rizvi, R. Pipetti, N. McIntyre, J. Todd, and I. Williams, "Threat model for securing internet of things (IoT) network at device-level," *Internet of Things*, vol. 11, p. 100240, 2020. https://doi.org/10.1016/j.iot.2020.100240

[16] A. K. Sahu, S. Sharma, M. Tanveer, and R. Raja, "Internet of Things attack detection using hybrid Deep Learning Model," *Comput Commun*, vol. 176, pp. 146–154, 2021. https://doi.org/10.1016/j.comcom.2021.05.024

[17] M. R. Islam and K. M. Aktheruzzaman, "An analysis of cybersecurity attacks against internet of things and security solutions," *Journal of Computer and Communications*, vol. 8, no. 04, p. 11, 2020.

[18] M. Aqeel, F. Ali, M. W. Iqbal, T. A. Rana, M. Arif, and M. R. Auwul, "A review of security and privacy concerns in the internet of things (IoT)," *J Sens*, vol. 2022, no. 1, p. 5724168, 2022. https://doi.org/10.1155/2022/5724168

[19] M. Shafiq, Z. Gu, O. Cheikhrouhou, W. Alhakami, and H. Hamam, "The Rise of 'Internet of Things': Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks," *Wirel Commun Mob Comput*, vol. 2022, no. 1, p. 8669348, 2022. https://doi.org/10.1155/2022/8669348

[20] G. P. Singh and P. K. Bangotra, "Internet of Things (IoT): vulnerability, attacks, and security," in *Wireless Sensor Networks and the Internet of Things*, Apple Academic Press, 2021, pp. 247–262.

[21] M. Wazzan, D. Algazzawi, O. Bamasaq, A. Albeshri, and L. Cheng, "Internet of Things botnet detection approaches: Analysis and recommendations for future research," *Applied Sciences*, vol. 11, no. 12, p. 5713, 2021.

[22] Y.-W. Chen, J.-P. Sheu, Y.-C. Kuo, and N. Van Cuong, "Design and implementation of IoT DDoS attacks detection system based on machine learning," in *2020 European Conference on Networks and Communications (EuCNC)*, IEEE, 2020, pp. 122–127. https://doi.org/10.1109/EuCNC48522.2020.9200909

[23] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad, and G. A. Shah, "IoT DoS and DDoS attack detection using ResNet," in *2020 IEEE 23rd International Multitopic Conference (INMIC)*, IEEE, 2020, pp. 1–6. https://doi.org/10.1109/INMIC50486.2020.9318216

[24] S. Evmorfos, G. Vlachodimitropoulos, N. Bakalos, and E. Gelenbe, "Neural network architectures for the detection of SYN flood attacks in IoT systems," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments*, 2020, pp. 1–4. https://doi.org/10.1145/3389189.3398000

[25] M. Roopak, G. Y. Tian, and J. Chambers, "An intrusion detection system against DDoS attacks in IoT networks," in *2020 10th annual computing and communication workshop and conference (CCWC)*, IEEE, 2020, pp. 562–567. https://doi.org/10.1109/CCWC47524.2020.9031206

[26] A. Dushimimana, T. Tao, R. Kindong, and A. Nishyirimbere, "Bi-directional recurrent neural network for intrusion detection system (IDS) in the internet of things (IoT)," *Int. J. Adv. Eng. Res. Sci*, vol. 7, pp. 524–539, 2020. https://dx.doi.org/10.22161/ijaers.73.68

[27] O. Malkawi, W. Almobaideen, N. Obaid, and B. Hammo, "Intrusion Detection System for 5G Device-to-Device Communication Technology in Internet of Things," *Informatica (Slovenia)*, vol. 48, no. 15, pp. 191–206, Oct. 2024, doi: 10.31449/inf.v48i15.4646.

[28] M. Almiani, A. AbuGhazleh, Y. Jararweh, and A. Razaque, "DDoS detection in 5G-enabled IoT networks using deep Kalman backpropagation neural network," *International Journal of Machine Learning and Cybernetics*, vol. 12, no. 11, pp. 3337–3349, 2021. https://doi.org/10.1007/s13042-021-01323-7

[29] F. F. Setiadi, M. W. A. Kesiman, and K. Y. E. Aryanto, "Detection of dos attacks using naive bayes method based on internet of things (iot)," in *Journal of Physics: Conference Series*, IOP Publishing, 2021, p. 012013. DOI: 10.1088/1742-6596/1810/1/012013

[30] A. Churcher *et al.*, "An experimental analysis of attack classification using machine learning in IoT networks," *Sensors*, vol. 21, no. 2, p. 446, 2021.

[31] S. Chesney, K. Roy, and S. Khorsandroo, "Machine learning algorithms for preventing IoT cybersecurity attacks," in *Intelligent Systems and Applications: Proceedings of the 2020 Intelligent Systems Conference (IntelliSys) Volume 3*, Springer, 2021, pp. 679–686. https://doi.org/10.1007/978-3-030-55190-2_53

[32] P. Singh Samom and A. Taggu, "Distributed denial of service (DDoS) attacks detection: A machine learning approach," in *Applied Soft Computing and Communication Networks: Proceedings of ACN 2020*, Springer, 2021, pp. 75–87. https://doi.org/10.1007/978-981-33-6173-7_6

[33] A. Izadi, N. Zarei, M. R. Nikoo, M. Al-Wardy, and F. Yazdandoost, "Exploring the potential of deep learning for streamflow forecasting: A comparative study with hydrological models for seasonal and perennial rivers," *Expert Syst Appl*, vol. 252, p. 124139, 2024. https://doi.org/10.1016/j.eswa.2024.124139

[34] H. Khajavi and A. Rastgoo, "Predicting the carbon dioxide emission caused by road transport using a Random Forest (RF) model combined by Meta-Heuristic Algorithms," *Sustain Cities Soc*, vol. 93, p. 104503, 2023. https://doi.org/10.1016/j.scs.2023.104503

[35] M. Ghadiri, A. A. Rassafi, and B. Mirbaha, "The effects of traffic zoning with regular geometric shapes on the precision of trip production models," *J Transp Geogr*, vol. 78, pp. 150–159, 2019. https://doi.org/10.1016/j.jtrangeo.2019.05.018

[36] S. Narkhede, "Understanding auc-roc curve," *Towards data science*, vol. 26, no. 1, pp. 220–227, 2018.

[37] H. Khajavi and A. Rastgoo, "Improving the prediction of heating energy consumed at residential buildings using a combination of support vector regression and meta-heuristic algorithms," *Energy*, vol. 272, p. 127069, 2023. https://doi.org/10.1016/j.energy.2023.127069

[38] A. Rastgoo and H. Khajavi, "A novel study on forecasting the airfoil self-noise, using a hybrid model based on the combination of CatBoost and Arithmetic Optimization Algorithm," *Expert Syst Appl*, vol. 229, p. 120576, 2023. https://doi.org/10.1016/j.eswa.2023.120576

[39] S. Gupta, G. Arango-Argoty, L. Zhang, A. Pruden, and P. Vikesland, "Identification of discriminatory antibiotic resistance genes among environmental resistomes using extremely randomized tree algorithm," *Microbiome*, vol. 7, pp. 1–15, 2019. https://doi.org/10.1186/s40168-019-0735-1

[40] E. Eslami, A. K. Salman, Y. Choi, A. Sayeed, and Y. Lops, "A data ensemble approach for real-time air quality forecasting using extremely randomized trees and deep neural networks," *Neural Comput Appl*, vol. 32, pp. 7563–7579, 2020. https://doi.org/10.1007/s00521-019-04287-6

[41] A. Sharaff and H. Gupta, "Extra-tree classifier with metaheuristics approach for email classification," in *Advances in Computer Communication and Computational Sciences: Proceedings of IC4S 2018*, Springer, 2019, pp. 189–197. https://doi.org/10.1007/978-981-13-6861-5_17

[42] U. Saeed, S. U. Jan, Y.-D. Lee, and I. Koo, "Fault diagnosis based on extremely randomized trees in wireless sensor networks," *Reliab Eng Syst Saf*, vol. 205, p. 107284, 2021. https://doi.org/10.1016/j.ress.2020.107284

[43] Ö. Akar and O. Güngör, "Rastgele orman algoritması kullanılarak çok bantlı görüntülerin sınıflandırılması," *Jeodezi ve Jeoinformasyon Dergisi*, no. 106, pp. 139–146, 2012. https://doi.org/10.9733/jgg.241212.1t

[44] A. Sarica, A. Cerasa, and A. Quattrone, "Random forest algorithm for the classification of neuroimaging data in Alzheimer's disease: a systematic review," *Front Aging Neurosci*, vol. 9, p. 329, 2017. https://doi.org/10.3389/fnagi.2017.00329

[45] M. Schonlau and R. Y. Zou, "The random forest algorithm for statistical learning," *Stata J*, vol. 20, no. 1, pp. 3–29, 2020. https://doi.org/10.1177/1536867X20909688

[46] K. VijiyaKumar, B. Lavanya, I. Nirmala, and S. S. Caroline, "Random forest algorithm for the prediction of diabetes," in *2019 IEEE international conference on system, computation, automation and networking (ICSCAN)*, IEEE,

2019, pp. 1–5. https://doi.org/10.1109/ICSCAN.2019.8878802

[47] E. K. Sahin, “Comparative analysis of gradient boosting algorithms for landslide susceptibility mapping,” *Geocarto Int*, vol. 37, no. 9, pp. 2441–2465, 2022. https://doi.org/10.1080/10106049.2020.1831623

[48] Y. Wan, M. Mao, L. Zhou, Q. Zhang, X. Xi, and C. Zheng, “A novel nature-inspired maximum power point tracking (MPPT) controller based on SSA-GWO algorithm for partially shaded photovoltaic systems,” *Electronics (Basel)*, vol. 8, no. 6, p. 680, 2019.

[49] Y. Yang, H. Chen, A. A. Heidari, and A. H. Gandomi, “Hunger games search: Visions, conception, implementation, deep analysis, perspectives, and towards performance shifts,” *Expert Syst Appl*, vol. 177, p. 114864, 2021. https://doi.org/10.1016/j.eswa.2021.114864

[50] H. Nguyen and X.-N. Bui, “A novel hunger games search optimization-based artificial neural network for predicting ground vibration intensity induced by mine blasting,” *Natural Resources Research*, vol. 30, no. 5, pp. 3865–3880, 2021. https://doi.org/10.1007/s11053-021-09903-8

[51] W. S. AbuShanab, M. Abd Elaziz, E. I. Ghandourah, E. B. Moustafa, and A. H. Elsheikh, “A new fine-tuned random vector functional link model using Hunger games search optimizer for modeling friction stir welding process of polymeric materials,” *journal of materials research and technology*, vol. 14, pp. 1482–1493, 2021. https://doi.org/10.1016/j.jmrt.2021.07.031

[52] S. Mirjalili, “Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm,” *Knowl Based Syst*, vol. 89, pp. 228–249, 2015. https://doi.org/10.1016/j.knosys.2015.07.006

[53] L. Yang, H. Nguyen, X.-N. Bui, T. Nguyen-Thoi, J. Zhou, and J. Huang, “Prediction of gas yield generated by energy recovery from municipal solid waste using deep neural network and moth-flame optimization algorithm,” *J Clean Prod*, vol. 311, p. 127672, 2021. https://doi.org/10.1016/j.jclepro.2021.127672

[54] O. Bozorg-Haddad, *Advanced optimization by nature-inspired algorithms*, vol. 720. Springer, 2018.

[55] S. H. S. Moosavi and V. K. Bardsiri, “Satin bowerbird optimizer: A new optimization algorithm to optimize ANFIS for software development effort estimation,” *Eng Appl Artif Intell*, vol. 60, pp. 1–15, 2017. https://doi.org/10.1016/j.engappai.2017.01.006

[56] S. Zhang, Y. Zhou, and Q. Luo, “A complex-valued encoding satin bowerbird optimization algorithm for global optimization,” *Evolving Systems*, vol. 12, pp. 191–205, 2021. https://doi.org/10.1007/s12530-019-09307-3

[57] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, “Slime mould algorithm: A new method for stochastic optimization,” *Future generation computer systems*, vol. 111, pp. 300–323, 2020. https://doi.org/10.1016/j.future.2020.03.055

[58] M. Abdel-Basset, R. Mohamed, R. K. Chakrabortty, M. J. Ryan, and S. Mirjalili, “An efficient binary slime mould algorithm integrated with a novel attacking-feeding strategy for feature selection,” *Comput Ind Eng*, vol. 153, p. 107078, 2021. https://doi.org/10.1016/j.cie.2020.107078

[59] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, and W. M. Mohamed, “Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems,” *Expert Syst Appl*, vol. 174, p. 114689, 2021. https://doi.org/10.1016/j.eswa.2021.114689

[60] Z. A. A. Alyasseri *et al.*, “Recent advances of whale optimization algorithm, its versions and applications,” *Handbook of Whale Optimization Algorithm*, pp. 9–31, 2024. https://doi.org/10.1016/B978-0-32-395365-8.00008-7

[61] A. G. Hussien, A. E. Hassanien, E. H. Houssein, M. Amin, and A. T. Azar, “New binary whale optimization algorithm for discrete optimization problems,” *Engineering Optimization*, vol. 52, no. 6, pp. 945–959, 2020. https://doi.org/10.1080/0305215X.2019.1624740

[62] N. Rana, M. S. A. Latiff, S. M. Abdulhamid, and H. Chiroma, “Whale optimization algorithm: a systematic review of contemporary applications, modifications and developments,” *Neural Comput Appl*, vol. 32, pp. 16245–16277, 2020. https://doi.org/10.1007/s00521-020-04849-z

[63] Q.-V. Pham, S. Mirjalili, N. Kumar, M. Alazab, and W.-J. Hwang, “Whale optimization algorithm with applications to resource allocation in wireless networks,” *IEEE Trans Veh Technol*, vol. 69, no. 4, pp. 4285–4297, 2020. https://doi.org/10.1109/TVT.2020.2973294

[64] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, “CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment,” *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[65] Z.-M. Gao and J. Zhao, “An improved grey wolf optimization algorithm with variable weights,” *Comput Intell Neurosci*, vol. 2019, no. 1, p. 2981282, 2019. https://doi.org/10.1155/2019/2981282