International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 5, 2025
https://theaspd.com/index.php

A Scalable XGB-RF Approach For Multi-Class Iot Botnet Detection

Preeti Suryawanshi[1], **Sonal Jagtap**[2,3]
[1]Research Scholar, Department of E&TC, Sinhgad College of Engineering, SPPU, Pune, India
[2]Research Guide, Department of E&TC, Sinhgad College of Engineering, SPPU, Pune, India,
[3]Professor, Department of E&TC, NBN Sinhgad Technical Institutes Campus, SPPU, Pune, India
[1]preeti37.phd@gmail.com [2,3] sonalkjagtap@gmail.com

Abstract - *IoT devices have changed very quickly in the last few years, which has made them more useful than ever for making life easier. But there are still a lot of security holes in IoT devices. This is mostly because most of them don't have the memory or processing power needed to support strong security features. Because of this, all kinds of cyberattacks can get to IoT devices. A single breach of a network system or device can put data security and privacy at risk. But you can use machine learning algorithms to find attacks on IoT devices. This work proposes a multimodal machine learning model called XGB-RF that can be used to find intrusion attacks. The N-BaIoT dataset, which includes harmful botnet attacks, was used with the proposed hybrid framework. The random forest (RF) algorithm was used to choose attributes, and the eXtreme Gradient Boosting (XGB) classifier was used to sort different kinds of attacks in IoT ecosystems. We use different performance measures to see how well the proposed XGB-RF model works. The results show that the model finds 99.94% of the attacks correctly. The proposed model does better on all the measures than the best algorithms that are currently available. The proposed method has a lot of potential to make IoT systems safer because it can find botnet attacks very well.*

Keywords: *IoT network security, botnet intrusion detection, feature selection, XGBoost, Random Forest (RF), Mirai, DDos.*

1. INTRODUCTION

The Internet of Things (IoT) is at the centre of the Fourth Industrial Revolution, which is being driven by new technologies [1]. Countries that are open to these new ideas will have a better chance of doing well in the future. IoT is one of the technologies that has grown the fastest in the history of technology. By the end of 2020, there were expected billions of devices connected to the Internet of Things [2]. But this rapid growth made security threats worse because IoT devices don't usually have strong security mechanisms, which make them easy targets for hackers. Every month, approximately 5,200 attacks happen on IoT devices [3]. In the first half of 2019, the number of attacks increased compared to the same time last year [4]. The report from Checkpoint showed that 71% of security experts noticed a rise in IoT threats after COVID-19 [5].

Targeted industries include blockchain supply chains [9,10], long-range (LoRa) networks [8], the Medical IoT [6,7], and smart industries [11]. Remarkably, billions of IoT devices have been infiltrated by the Mirai and BASHLITE botnets, which were identified in 2016 and 2015, respectively [12]. Distributed Denial-of-Service (DDoS) was a most common IoT threat assault based on Mirai in 2018. Furthermore, it is projected that by 2023, there would be 15.4 million DDoS attacks worldwide [13]. Developing strong attack detection techniques is crucial in light of this changing threat scenario [14]. Conventional security measures are insufficient to prevent botnet assaults based on the Internet of Things.

Researchers have made increasing progress in addressing this by developing Intrusion Detection Systems (IDS), especially for the two main forms of IoT-based attacks: network-based [16] and host-based [15]. Support Vector Machine (SVM), K-Nearest Neighbor (KNN), neural networks, and Naïve Bayes are examples of traditional machine learning (ML) methods for intrusion detection [17–19]. In their investigation, Shafiq et al. [20] identified Naïve Bayes as the most successful machine learning technique for detecting harmful IoT activities. The N-BaIoT dataset was also subjected to feature selection by Soe et al. [21] using J48 decision trees, Naïve Bayes, and Artificial Neural Networks (ANN), whose accuracy was over 99%. Diro et al. [22] also presented a deep learning-based intrusion detection system that exhibited much higher detection rates.

The NSL-KDD dataset was examined using Random Forest (RF), SVM, and Extreme Learning Machine (ELM) by Ahmad et al. [23], who discovered that ELM performed better than SVM in large datasets. With a detection rate of 96.8%, Deng et al. [24] created a network-based intrusion detection system based on K-means clustering and manual feature extraction. Mirsky et al. [25] used ensemble autoencoders for offline training and online deployment to create a lightweight, unsupervised IDS model. Using the publicly available CIC-IDS 2017 dataset, Radford et al. [26] proposed another unsupervised learning method for anomaly detection with Long Short-Term Memory (LSTM) recurrent neural networks. However, the performance of machine learning techniques for host-based intrusion detection in Internet of Things (IoT) networks has not been extensively studied in the majority of literatures [27].

The literature indicates that ML-based technology is very effective in security of IoT. Since most attacks occur in real time, the detection model must be accurate and efficient. To ensure this, a reduction in dimensionality of features would be used to reduce complexity and increase execution speed, the most important for real-time intrusion detection. To meet this requirement, we introduce machine learning solutions based on N-BaIoT data sets. Initially, we use random forest algorithms (RF) to rank the most relevant features to improve detection accuracy. Then we identify and label malicious activities using the eXtreme Gradient Boosting (XGB) algorithm. This combination is especially suitable for large data scenarios where speed and accuracy are important. Unlike Recursive Feature Elimination (RFE) and RFECV, which add individual features and drop those that have little impact on classification, RF ranks feature according to information gain and provides a less difficult selection process. The model is compared to some of the most advanced ML algorithms on the N-BaIoT dataset and outperforms them consistently

Main Contributions of the Paper:

- A new hybrid intrusion detection model, XGB-RF, involving the use of RF as a feature selector and XGB as the ultimate classifier.
- In-depth comparison with other advanced ML models, outperforming in every way.

2. MATERIALS AND METHODS

The hybrid mechanism adopted by N-BaIoT multi-class. Attack detection is known as XGB-RF classification. RF feature selection algorithms are initially applied in the selection of functional features. XGB classification algorithm comes next is employed to detect any type of attack on the IoT network. Due to this, our proposed approach is referred to as XGB-RF. The multiplier of NBaIoT is reduced from 115 to 40 functions applicable to the same window by our technique.The system is presented below (Figure 1).

2.1 Acquisition of data

The study used the N-BaIoT dataset (28), published in 2020, which contains traffic from nine IoT sensors captured on the local network using wireshark on the central switch (seehttps://www.kaggle.com/mkashifn/nbaiot-dataset, accessed June 5, 2021). It consists of 115 statistical features derived from package capture files (pcap). These are calculated using seven statistical metrics on five-time windows (100ms, 500ms, 1.5ms, 10ms, and 1ms): average, variance, number, size, radius, covariance, and correlation coefficients. This structure allows for data sets for state-level intrusion detection systems (IDS) to analyze traffic in a fixed temporal context.

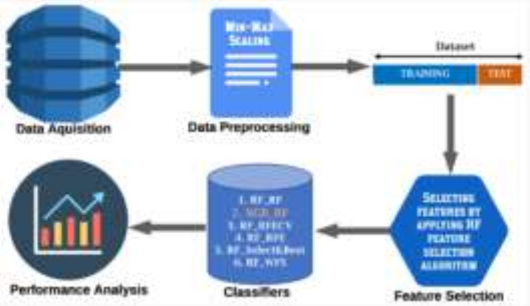


Fig 1. Workflow diagram of the proposed system.

The dataset includes 229,829 total samples, with 13,113 benign and 216,716 malicious samples. The malicious data spans 10 distinct attack types (plus benign traffic for a total of 11 classes):

- Class_1: Benign – normal traffic with no malicious behavior.
- Class_2 & Class_4: Mirai ACK and Mirai SYN – flooding attacks that overwhelm servers by exploiting the TCP handshake process.
- Class_3: Mirai Scan – scans for vulnerable devices using telnet (port 23 or 2223) and connects them to a command-and-control (C&C) server.
- Class_5: Mirai UDP – randomized UDP flooding using varied source/destination ports and IPs to evade detection.
- Class_6: Mirai UDP Plain – a more targeted version of the UDP flood that focuses on fewer high-use ports.
- Class_7 to Class_11: Gafgyt Combo, Junk, Scan, TCP, UDP – attacks from the Gafgyt botnet family, capable of DDoS, vulnerability scanning, malware execution, and C&C operations. Gafgyt mainly targets smart routers using weak credentials or known vulnerabilities.

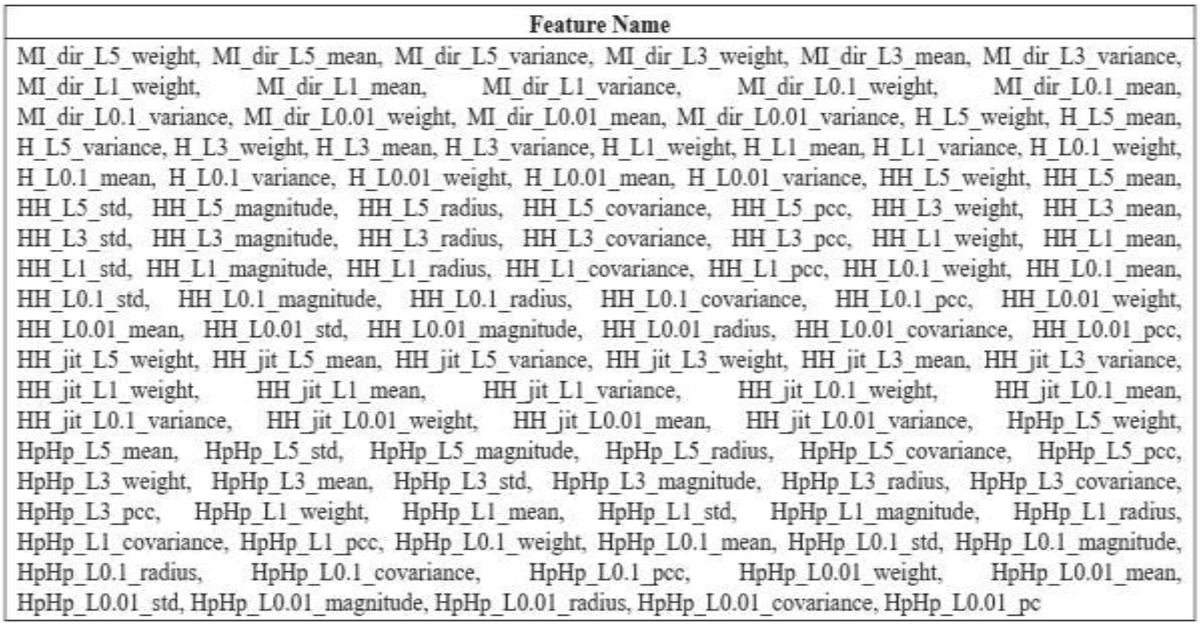
2.2 Data Pre-Processing

The N-BaIoT dataset is highly imbalanced, with benign records significantly fewer than attack samples. Each record includes statistically derived features labeled in the format: <Header>_<TimeWindow>_<StatisticalMeasure>.These capture data such as:

- Header information (e.g., jitter, packet count)
- Five separate time windows
- Seven statistical metrics

From four primary metrics—packet count, jitter, outbound packet size, and combined in/out packet size—23 base features are computed. These are expanded across five-time windows, resulting in 115 total features. This multi-window, multi-statistic format ensures the dataset captures both short-term and long-term behavior patterns, making it ideal for IDS models. The study uses these features as input for feature selection (RF) and classification (XGB). The full list of 115 features is shown in Table 1, with corresponding abbreviations explained in Table 2.

Table 1. Name of the features used in this study



Feature Name
MI_dir_L5_weight, MI_dir_L5_mean, MI_dir_L5_variance, MI_dir_L3_weight, MI_dir_L3_mean, MI_dir_L3_variance, MI_dir_L1_weight, MI_dir_L1_mean, MI_dir_L1_variance, MI_dir_L0.1_weight, MI_dir_L0.1_mean, MI_dir_L0.1_variance, MI_dir_L0.01_weight, MI_dir_L0.01_mean, MI_dir_L0.01_variance, H_L5_weight, H_L5_mean, H_L5_variance, H_L3_weight, H_L3_mean, H_L3_variance, H_L1_weight, H_L1_mean, H_L1_variance, H_L0.1_weight, H_L0.1_mean, H_L0.1_variance, H_L0.01_weight, H_L0.01_mean, H_L0.01_variance, HH_L5_weight, HH_L5_mean, HH_L5_std, HH_L5_magnitude, HH_L5_radius, HH_L5_covariance, HH_L5_pcc, HH_L3_weight, HH_L3_mean, HH_L3_std, HH_L3_magnitude, HH_L3_radius, HH_L3_covariance, HH_L3_pcc, HH_L1_weight, HH_L1_mean, HH_L1_std, HH_L1_magnitude, HH_L1_radius, HH_L1_covariance, HH_L1_pcc, HH_L0.1_weight, HH_L0.1_mean, HH_L0.1_std, HH_L0.1_magnitude, HH_L0.1_radius, HH_L0.1_covariance, HH_L0.1_pcc, HH_L0.01_weight, HH_L0.01_mean, HH_L0.01_std, HH_L0.01_magnitude, HH_L0.01_radius, HH_L0.01_covariance, HH_L0.01_pcc, HH_jit_L5_weight, HH_jit_L5_mean, HH_jit_L5_variance, HH_jit_L3_weight, HH_jit_L3_mean, HH_jit_L3_variance, HH_jit_L1_weight, HH_jit_L1_mean, HH_jit_L1_variance, HH_jit_L0.1_weight, HH_jit_L0.1_mean, HH_jit_L0.1_variance, HH_jit_L0.01_weight, HH_jit_L0.01_mean, HH_jit_L0.01_variance, HpHp_L5_weight, HpHp_L5_mean, HpHp_L5_std, HpHp_L5_magnitude, HpHp_L5_radius, HpHp_L5_covariance, HpHp_L5_pcc, HpHp_L3_weight, HpHp_L3_mean, HpHp_L3_std, HpHp_L3_magnitude, HpHp_L3_radius, HpHp_L3_covariance, HpHp_L3_pcc, HpHp_L1_weight, HpHp_L1_mean, HpHp_L1_std, HpHp_L1_magnitude, HpHp_L1_radius, HpHp_L1_covariance, HpHp_L1_pcc, HpHp_L0.1_weight, HpHp_L0.1_mean, HpHp_L0.1_std, HpHp_L0.1_magnitude, HpHp_L0.1_radius, HpHp_L0.1_covariance, HpHp_L0.1_pcc, HpHp_L0.01_weight, HpHp_L0.01_mean, HpHp_L0.01_std, HpHp_L0.01_magnitude, HpHp_L0.01_radius, HpHp_L0.01_covariance, HpHp_L0.01_pc

2.3 Feature Selection

To enhance the classification performance and minimize model complexity in IoT-based Intrusion Detection Systems (IDS), four commonly known feature selection methods are used: Recursive Feature Elimination (RFE), Recursive Feature Elimination Using Cross-Validation (RFECV), SelectK-Best, and Random Forest (RF)-based feature importance. A concise overview of these techniques is provided below.

2.3.1 RF-Based Feature Selection

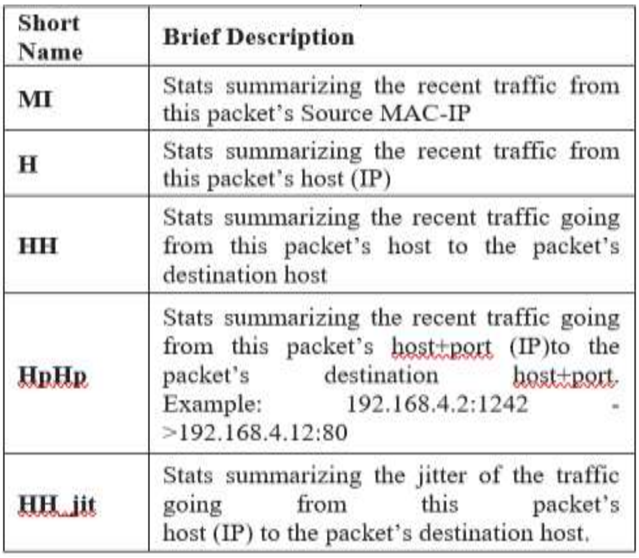
Random Forests (RF) can assess the relative importance of each feature based on information gain, allowing for efficient feature selection. The detailed implementation of RF-based feature ranking is elaborated in Section 2.4.1.

2.3.2 Recursive Feature Elimination (RFE)

RFE is a wrapper-based feature selection method that

identifies the most relevant features by recursively removing the least significant ones. This technique is particularly effective in reducing redundancy in high-dimensional datasets, thus enhancing classifier performance and reducing

Table 2. Feature description.



Short Name	Brief Description
MI	Stats summarizing the recent traffic from this packet's Source MAC-IP
H	Stats summarizing the recent traffic from this packet's host (IP)
HH	Stats summarizing the recent traffic going from this packet's host to the packet's destination host
HpHp	Stats summarizing the recent traffic going from this packet's host+port (IP)to the packet's destination host+port. Example: 192.168.4.2:1242 ->192.168.4.12:80
HH_jit	Stats summarizing the jitter of the traffic going from this packet's host (IP) to the packet's destination host.

redundancy in high-dimensional datasets, thus enhancing classifier performance and reducing bias. In this study, RFE is applied in a stepwise fashion to rank features based on their importance using an RF classifier. The process iteratively eliminates the lowest-ranked features, retrains the model on the remaining subset, and evaluates the performance at each step. This continues until the desired number of optimal features is selected or all features have been assessed.

Following Algorithm illustrates the procedural steps of RFE, Recursive Feature Elimination (RFE)

- Initialization:

Begin with the full set of features C and a classifier. Evaluate the model performance using all features.

- Pre-process the input dataset
- N=len (C) where C= Total number of features
- for i=1 to N do

```
  a. Train the classifier using the current set of features
  b. Compute feature importances
  c. Eliminate the less important feature
  d. Evaluate  model performance with the remaining feature subset
end
```

- Result: Feature subset corresponding to the best model performance during iterations while Figure 2 gives the iterative feature elimination workflow visual representation. Ultimately, the goal is to obtain a feature subset that maximizes classification accuracy with minimal complexity

2.3.3 Recursive Feature Elimination with Cross-Validation (RFECV)

Recursive Feature Elimination with Cross Validation (RFECV) is a strong algorithm that assists to remove and retain only the most important features from big data sets. Similar to RFE, it eliminates the less important features one at a time, but with a significant difference—RFECV employs cross validation at every step to evaluate how well the existing model fits with fresh (untrained) data. This automatically identifies the best features with the top cross-validation scores, providing more reliable sets of features to the estimate given.

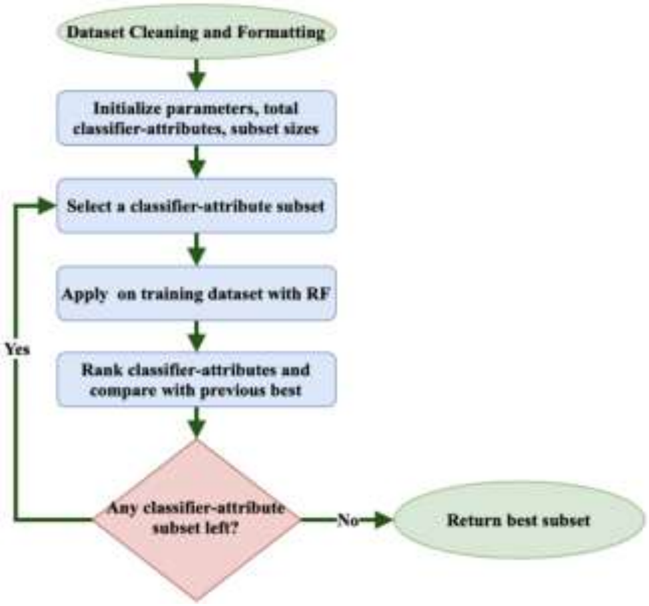

Dataset Cleaning and Formatting
Initialize parameters, total classifier-attributes, subset sizes
Select a classifier-attribute subset
Apply on training dataset with RF
Rank classifier-attributes and compare with previous best
Any classifier-attribute subset left?
Yes
No
Return best subset

Fig 2. Workflow diagram of Recursive Feature Elimination (RFE).

2.3.3 Recursive Feature Elimination with Cross-Validation (RFECV)

Recursive Feature Elimination with Cross Validation (RFECV) is a strong algorithm that assists to remove and retain only the most important features from big data sets. Similar to RFE, it eliminates the less important features one at a time, but with a significant difference—RFECV employs cross validation at every step to evaluate how well the existing model fits with fresh (untrained) data. This automatically identifies the best features with the top cross-validation scores, providing more reliable sets of features to the estimate given.

2.3.4 Select-K-Best

Select-K-Best is a filter-based method that combines statistical tests and selects the highest K-value features of all input features. Each feature is evaluated independently (one-by-one) by metric methods such as chi-squared, mutual information value or ANOVA F value, or mutual information, depending on the type of task, classifying or refining. This approach is computationally efficient, especially useful when it is necessary to reduce the speed and interpretation of features before modelling.

2.1 Classifiers

This section outlines the core classification algorithms utilized in the proposed system, emphasizing their interpretability, performance, and suitability for IoT-based intrusion detection.

2.4.1 Random Forest (RF)

Random Forest (RF) is a group learning algorithm introduced by L. Breiman in 2001 based on the Bagging (Bootstrap Aggregating) technique. The "forest" of the decision tree is constructed, each of which is trained on a randomly selected data subset (with replacement) and the final prediction is based on a majority in all trees. This ensemble approach greatly reduces overfitting and variance and makes RF very robust, especially for large and high-dimensional data sets. Beyond classification, RF is widely used to select features, leveraging the intrinsic ability of decision trees to evaluate the feature importance. According to Genuer et al. [31], the importance of variables can be effectively measured by various indices such as Gini impurities, average accuracy decline, and permutation importance, each offering different perspectives on the importance of a characteristic for the predictive capability of the model. RF's strengths lie in:

- Handling missing or noisy data,
- Capturing nonlinear relationships,
- Managing multicollinearity,
- Providing interpretable feature importance rankings.

In this study, Random Forest is classifier as well as feature selection tool. Based on the computed importance scores, 40 key features are used for classification from the original 115-feature N-BaIoT dataset. The final classification outcome is aggregated from individual tree predictions, as expressed in Equation (1).

$$C(t) = \max_{p} 1 \ E_t\left[\sum_{i=1}^{K}(c_i(T) = P)\right] \qquad (1)$$

The proposed method takes the training set T from the original dataset S and is further divided into subsets K. Each subset generates a K decision tree using random vectors. The classification result is C(t), and ci(T) represents the output of the ith decision tree and P represents the target class.

Random Forest uses several hyperparameters to improve model accuracy and computational efficiency. Because of its inherent ability to handle high-dimensional data, RF effectively performs a functional selection process (FS). The importance of characteristics in the RF is usually quantified by Gini importance, which measures the contribution of each characteristic to the model.The Gini impurity at a node t is given by:

$$i(t) = 1 - f_1^2 - f_0^2 \qquad (2)$$

where fj is the fraction of samples of class j∈{0,1}, calculated as:

$$f_j = \frac{n_j}{n} \qquad (3)$$

Here, nj is the sample count in class j, and n is the total count of sample at node t. To evaluate feature splits, the decrease in impurity δ_i from a split is computed as:

$$\delta i(t) = i(t) - f_p^i i(t_p) - f_q^i i(t_q) \qquad (4)$$

where tp and tq are child nodes resulting from a split on variable Θ, and fp, fq are the respective proportions of samples sent to each child. An exhaustive search over all potential split thresholds Θ is conducted, and the reduction in Gini impurity is aggregated for each feature using:

$$I_G(\Theta) = \sum_r \sum_q \delta i_\Theta(t, T) \qquad (5)$$

This cumulative score indicates how often a feature contributes to significant splits, guiding the selection of the most relevant attributes for the classifier.

2.4.2. eXtreme Gradient Boosting (XGBoost)

Following feature selection via RF, XGBoost (XGB) is employed for botnet attack detection using the selected features. XGBoost is a robust ensemble technique that improves model performance through gradient-boosted decision trees [32]. Unlike RF, which builds independent trees, XGB constructs trees sequentially—each tree learns from the residuals of its predecessors, enhancing the predictive power iteratively [33].

Each boosting round minimizes the residuals—i.e., the difference between actual values and predicted values. Once the residual falls below a predefined threshold, the training process concludes. Alternatively, if a set number of trees is reached before convergence, the most recent model is finalized

Key strengths of XGBoost include support for regularization, parallel processing, and efficient computation, making it faster and more scalable than standard gradient boosting methods. XGBoost's performance is evaluated using the objective function:

$P(\theta)=t(\theta)+r(\theta)$ (6)

where θ denotes model parameters, t(θ) is the training loss, and r(θ) represents the regularization term [34].

2.2 Performance Analysis 0f Model

This study focuses on developing a classification model using training data sets and rigorously assessing their performance across the whole data set. To ensure a complete assessment, several evaluation metrics are used: accuracy (ACC), F1 scores, Kappa indexes, Matthew correlation coefficients (MCC), sensitivity (SE), specificity (SP), threat scores, and balance accuracy.

These metrics collectively capture the classifier's effectiveness across different performance dimensions—including overall correctness, class balance handling, and robustness to imbalanced data. Additionally, Section 3.2 presents six confusion matrices, each corresponding to a different classifier, to further illustrate and compare classification performance in a visual and interpretable format.

3. EXPERIMENTAL RESULTS

Experimental comparison on the raw test data is performed using a hold-out validation strategy with 75% used for training and 25% for testing. Comparison of the performance of the classification model and the corresponding confusion matrices is shown in Sections 3.1 and 3.2. Additional information, including performance at varying train-test ratios, ROC curve analysis, and comparative benchmarking against the existing literature, is shown in Sections 3.3, 3.4, and 3.5.

3.1. Performance Measures

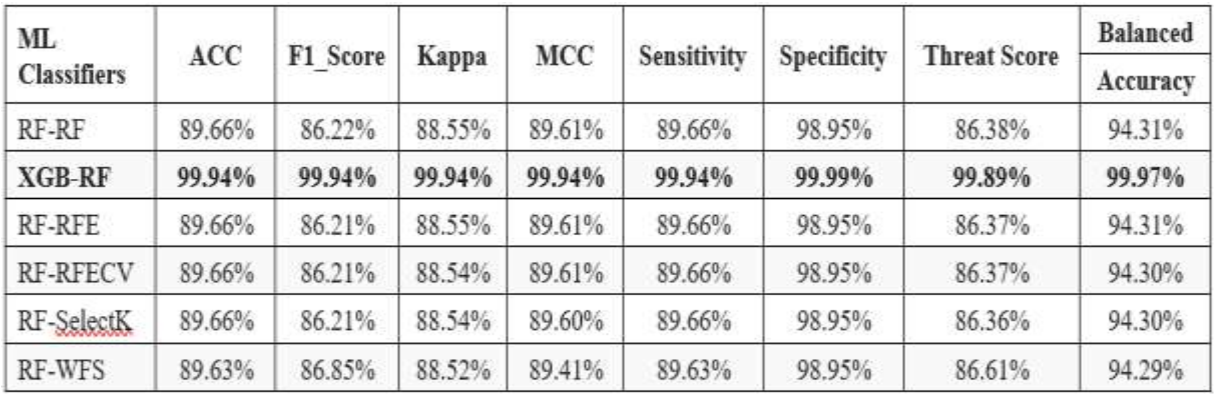
After applying Recursive Feature Elimination (RFE) with Random Forest (RF) and classifying the selected features using XGBoost (XGB), we assess the performance using various statistical metrics. To benchmark the

effectiveness of the proposed method, five additional ML models are evaluating

- RF-RF: RF classifier with RF-based feature selection
- RF-RFE: RF classifier with RFE
- RF-RFECV: RF classifier with RFECV
- RF-SelectK: RF classifier with Select-K-best
- RF-WFS: RF classifier without any feature selection

Table3 Summarize classification performance of all models

Table 3. Performance of Classification Algorithms.



ML Classifiers	ACC	F1_Score	Kappa	MCC	Sensitivity	Specificity	Threat Score	Balanced Accuracy
RF-RF	89.66%	86.22%	88.55%	89.61%	89.66%	98.95%	86.38%	94.31%
XGB-RF	**99.94%**	**99.94%**	**99.94%**	**99.94%**	**99.94%**	**99.99%**	**99.89%**	**99.97%**
RF-RFE	89.66%	86.21%	88.55%	89.61%	89.66%	98.95%	86.37%	94.31%
RF-RFECV	89.66%	86.21%	88.54%	89.61%	89.66%	98.95%	86.37%	94.30%
RF-SelectK	89.66%	86.21%	88.54%	89.60%	89.66%	98.95%	86.36%	94.30%
RF-WFS	89.63%	86.85%	88.52%	89.41%	89.63%	98.95%	86.61%	94.29%

Among these, the XGB-RF model demonstrated superior performance, achieving the highest accuracy (99.9426%), along with top scores in sensitivity, specificity, F1 score, and balanced accuracy (99.9683%).

It also recorded the lowest error rate (0.06%), significantly outperforming the other models. Conversely, the RF-WFS approach yielded the least favorable results.

Additionally, execution efficiency was evaluated: on a system with an Intel Core i9 processor and 64 GB RAM, the XGB-RF model processed 57,458 test instances time of 0.0010063 seconds per instance—indicating its suitability for real-time botnet detection.

3.2 Evaluation on Different Train-Test Schemes

While our proposed model primarily used a 75–25% train-test split, informed by prior experience, we also evaluated its robustness using 70–30% and 67–33% splits—approaches commonly used in N-BaIoT studies [35, 36]. As shown in Table 4, the variation in data split had minimal
impact on performance, reaffirming the model's stability across different data distributions.

3.3 Confusion Matrix

To analyze classification performance in a multi-class setting, confusion matrices are employed, particularly suited for datasets like N-BaIoT, which contain multiple attack categories. These matrices help visualize misclassifications and class-wise prediction accuracy. The confusion matrices for all evaluated classifiers are presented in Figure 3, offering deeper insight into the decision boundaries and prediction consistency of each model

Table 4. Performance Evaluation on Different Train-Test Schemes

Performance	Train-Test (70–30%)	Train-Test (67–33%)
Accuracy	99.96%	99.95%
F1_Score	99.96%	99.95%
Kappa	99.95%	99.95%
MCC	99.95%	99.95%
Sensitivity	99.96%	99.95%
Specificity	100.00%	100.00%
Threat_Score	99.92%	99.92%
Balanced Accuracy	99.98%	99.97%

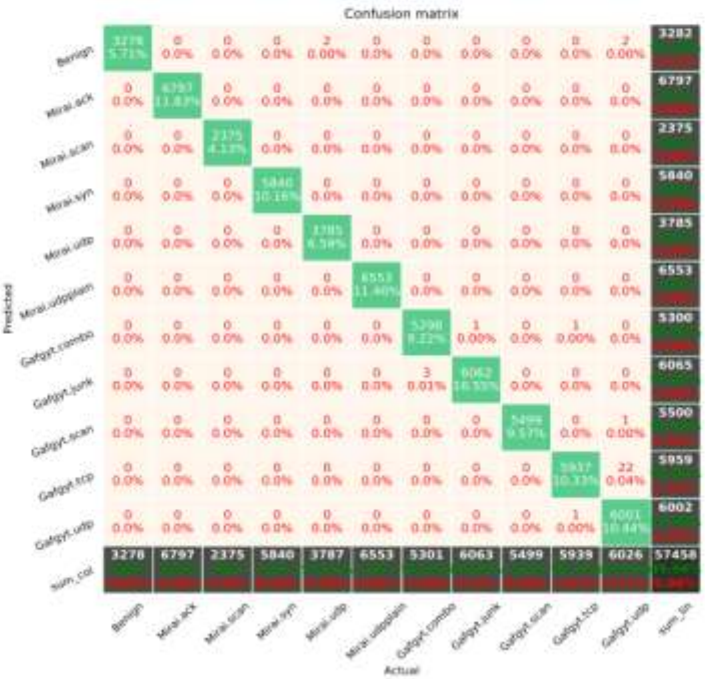
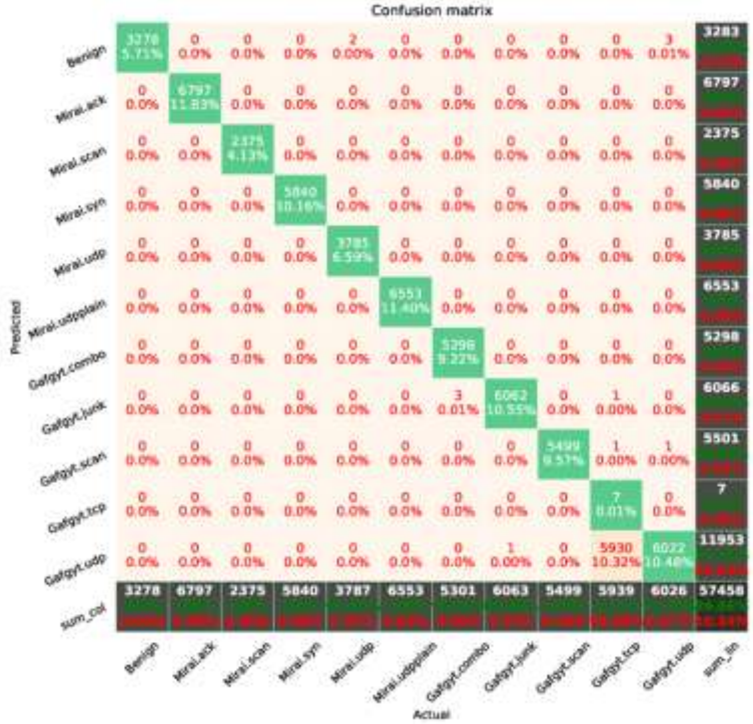


Fig 3a. Confusion matrix for (a) RF-RF

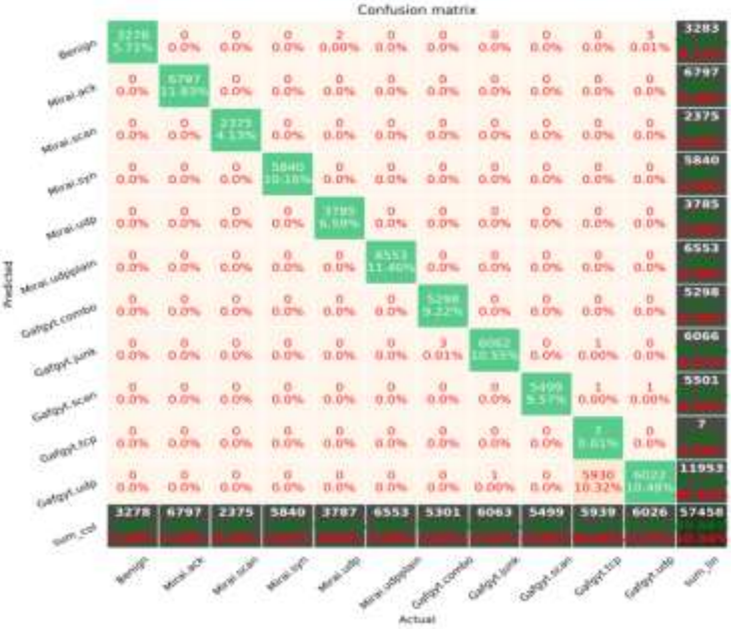
3.4. ROC Curve Analysis

As observed in Figure 3, the proposed XGB-RF model effectively classifies all attack types, including Gafgyt.udp, which is consistently misclassified by competing all other 5 different methods discussed in fig 3a,3b,3c,3d,3e through confusion Matrix. To further analyze this, we generated an ROC curve specifically for Gafgyt.udp vs. all other classes, as shown in Figure 4. The proposed model demonstrates a near-perfect AUROC ≈ 1.0, indicating outstanding discriminative power



	Benign	Mirai.ack	Mirai.scan	Mirai.syn	Mirai.udp	Mirai.udpplain	Gafgyt.combo	Gafgyt.junk	Gafgyt.scan	Gafgyt.tcp	Gafgyt.udp	sum_lin
Benign	3278 5.71%	0 0.0%	0 0.0%	0 0.0%	2 0.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.01%	3283
Mirai.ack	0 0.0%	6797 11.83%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6797
Mirai.scan	0 0.0%	0 0.0%	2375 4.13%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2375
Mirai.syn	0 0.0%	0 0.0%	0 0.0%	5840 10.16%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5840
Mirai.udp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3785 6.59%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3785
Mirai.udpplain	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6553 11.40%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6553
Gafgyt.combo	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5298 9.22%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5298
Gafgyt.junk	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.01%	6062 10.55%	0 0.0%	1 0.00%	0 0.0%	6066
Gafgyt.scan	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5499 9.57%	1 0.00%	1 0.00%	5501
Gafgyt.tcp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.01%	0 0.0%	7
Gafgyt.udp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.00%	0 0.0%	5930 10.32%	6022 10.48%	11953
sum_col	3278	6797	2375	5840	3787	6553	5301	6063	5499	5939	6026	57458

Fig 3b. Confusion matrix for XGB-RF



	Benign	Mirai.ack	Mirai.scan	Mirai.syn	Mirai.udp	Mirai.udpplain	Gafgyt.combo	Gafgyt.junk	Gafgyt.scan	Gafgyt.tcp	Gafgyt.udp	sum_lin
Benign	3278 5.71%	0 0.0%	0 0.0%	0 0.0%	2 0.00%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.01%	3283
Mirai.ack	0 0.0%	6797 11.83%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6797
Mirai.scan	0 0.0%	0 0.0%	2375 4.13%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	2375
Mirai.syn	0 0.0%	0 0.0%	0 0.0%	5840 10.16%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5840
Mirai.udp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3785 6.59%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3785
Mirai.udpplain	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6553 11.40%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	6553
Gafgyt.combo	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5298 9.22%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5298
Gafgyt.junk	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	3 0.01%	6062 10.55%	0 0.0%	1 0.00%	0 0.0%	6066
Gafgyt.scan	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	5499 9.57%	1 0.00%	1 0.00%	5501
Gafgyt.tcp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	7 0.01%	0 0.0%	7
Gafgyt.udp	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.00%	0 0.0%	5930 10.32%	6022 10.48%	11953
sum_col	3278	6797	2375	5840	3787	6553	5301	6063	5499	5939	6026	57458

Fig 3c. Confusion matrix for RF-RFECV

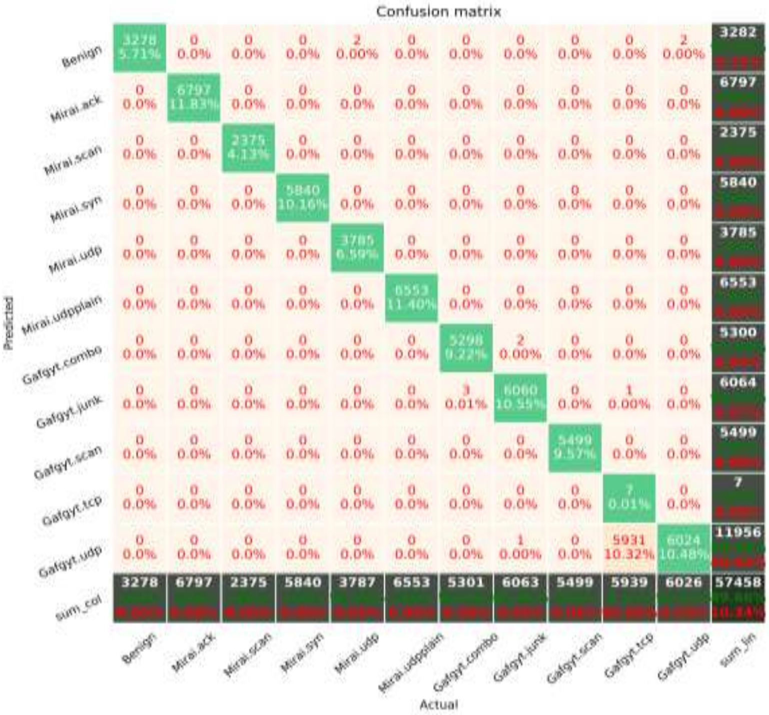


Fig 3d. Confusion matrix for RF-RFE

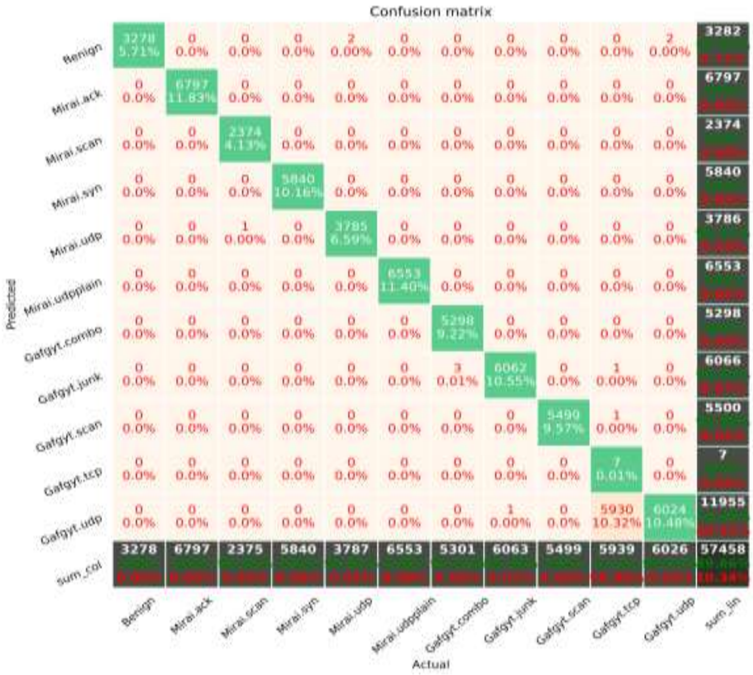


Fig 3e. Confusion matrix for RF-Select

In contrast, the ROC curves for the alternative methods almost completely overlap, suggesting similarly poor performance in this classification task. According to Mason et al. [37], an AUROC of ~1.0 aligns with Mann–Whitney U-statistics and reflects statistical significance ($p < 0.001$). Given this, additional significance testing such as ANOVA was deemed unnecessary.

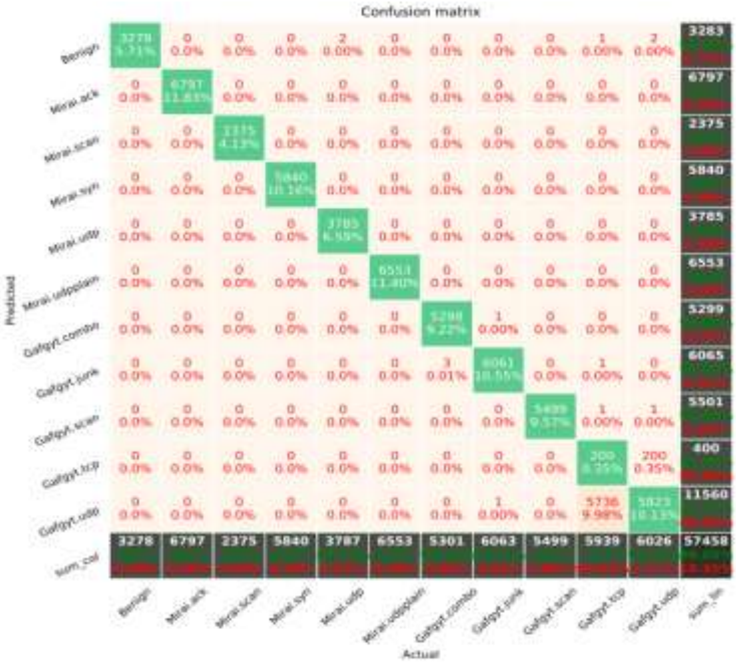
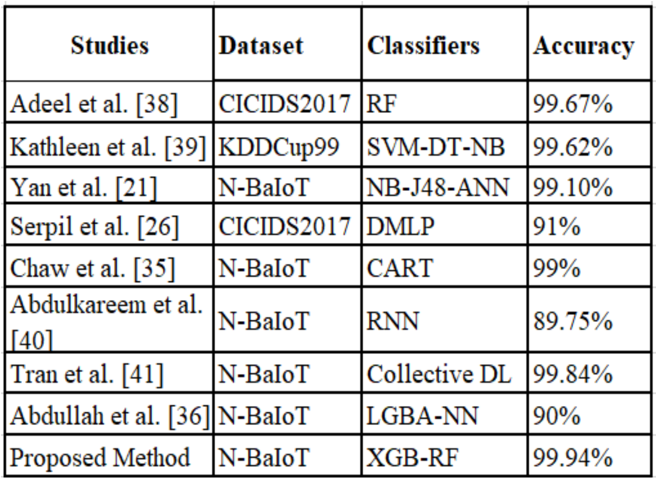


Figure 3. Confusion matrix for RF-WFS

3.5. Validation of Performance with Previous work

To validate our model's generalizability, we compared it against several existing approaches (see Table 5). Notably, studies using the same N-BaIoT dataset include:

Table 5. Comparative performance analysis of proposed method



Studies	Dataset	Classifiers	Accuracy
Adeel et al. [38]	CICIDS2017	RF	99.67%
Kathleen et al. [39]	KDDCup99	SVM-DT-NB	99.62%
Yan et al. [21]	N-BaIoT	NB-J48-ANN	99.10%
Serpil et al. [26]	CICIDS2017	DMLP	91%
Chaw et al. [35]	N-BaIoT	CART	99%
Abdulkareem et al. [40]	N-BaIoT	RNN	89.75%
Tran et al. [41]	N-BaIoT	Collective DL	99.84%
Abdullah et al. [36]	N-BaIoT	LGBA-NN	90%
Proposed Method	N-BaIoT	XGB-RF	99.94%

4. Conclusions

As IoT continues to shape the fabric of our economic and societal systems, safeguarding its infrastructure becomes increasingly critical. Intrusion Detection Systems (IDS) remain a cornerstone of IoT security.

This study proposed a hybrid XGB-RF machine learning system for efficient detection of IoT botnet attacks using the N-BaIoT dataset. Among five comparative schemes, the XGB-RF model consistently achieved superior results—boasting accuracy and sensitivity over 99%, with a 10% margin over baseline models. It achieved single-instance detection in just 0.0010063 seconds, supporting its real-time applicability.

While the model excels in detecting known threats, identifying zero-day or unknown attacks remains a challenge. In future work, we aim to reduce detection latency further and explore generalization to unknown attacks, making the system even more adaptive to evolving IoT threat landscapes.

Conflicts of Interest

The authors state that there are no conflicts of interest related to the publication of this paper.

Funding Statement

This study was not supported by any particular grant from funding organizations in the public, commercial, or nonprofit sectors.

Acknowledgments

The authors wish to convey their heartfelt thanks to STES Sinhgad College of Engineering for offering the essential resources and assistance to conduct this research. Heartfelt gratitude to the research community whose contributions have established the groundwork for this study.

REFERENCES

[1] Fallahpour, A.; Wong, K.Y.; Rajoo, S.; Fathollahi-Fard, A.M.; Antucheviciene, J.; Nayeri, S. An integrated approach for a sustainable supplier selection based on Industry 4.0 concept. Environ. Sci. Pollut. Res. 2021, 1–19. [Google Scholar] [CrossRef] [PubMed]

[2] Attaran, M. The internet of things: Limitless opportunities for business and society. J. Strateg. Innov. Sustain. 2017, 12, 11–29. [Google Scholar]

[3] Symantec Internet Security Threat Report. 2019. Available online: https://docs.broadcom.com/doc/istr-24-2019-en (accessed on 30 June 2021).

[4] Fruhlinger, J. Top Cybersecurity Facts, Figures and Statistics. 2020. Available online: https://www.csoonline.com/article/3153707/top-cybersecurity-facts-figures-and-statistics.html (accessed on 30 June 2021).

[5] A Perfect Storm: The Security Challenges of Coronavirus Threats and Mass Remote Working. 2020. Available online: https://blog.checkpoint.com/2020/04/07/a-perfect-storm-the-security-challenges-of-coronavirus-threats-and-mass-remote-working/ (accessed on 30 June 2021).

[6] Manimurugan, S.; Al-Mutairi, S.; Aborokbah, M.M.; Chilamkurti, N.; Ganesan, S.; Patan, R. Effective attack detection in internet of medical things smart environment using a deep belief neural network. IEEE Access 2020, 8, 77396–77404. [Google Scholar] [CrossRef]

[7] Fathollahi-Fard, A.M.; Ahmadi, A.; Karimi, B. Multi-Objective Optimization of Home Healthcare with Working-Time Balancing and Care Continuity. Sustainability 2021, 13, 12431. [Google Scholar] [CrossRef]

[8] Muthanna, M.S.A.; Muthanna, A.; Rafiq, A.; Hammoudeh, M.; Alkanhel, R.; Lynch, S.; Abd El-Latif, A.A. Deep reinforcement learning based transmission policy enforcement and multi-hop routing in QoS aware LoRa IoT networks. Comput. Commun. 2021, 183, 33–50. [Google Scholar] [CrossRef]

[9] Fathollahi-Fard, A.M.; Dulebenets, M.A.; Hajiaghaei-Keshteli, M.; Tavakkoli-Moghaddam, R.; Safaeian, M.; Mirzahosseinian, H. Two hybrid meta-heuristic algorithms for a dual-channel closed-loop supply chain network design problem in the tire industry under uncertainty. Adv. Eng. Inform. 2021, 50, 101418. [Google Scholar] [CrossRef]

[10] Moosavi, J.; Naeni, L.M.; Fathollahi-Fard, A.M.; Fiore, U. Blockchain in supply chain management: A review, bibliometric, and network analysis. Environ. Sci. Pollut. Res. 2021, 5, 1–15. [Google Scholar] [CrossRef]

[11] Rafiq, A.; Ping, W.; Min, W.; Muthanna, M.S.A. Fog Assisted 6TiSCH Tri-Layer Network Architecture for Adaptive Scheduling and Energy-Efficient Offloading Using Rank-Based Q-Learning in Smart Industries. IEEE Sens. J. 2021, 21, 25489–25507. [Google Scholar] [CrossRef]

[12] Marzano, A.; Alexander, D.; Fonseca, O.; Fazzion, E.; Hoepers, C.; Steding-Jessen, K.; Chaves, M.H.; Cunha, Í.; Guedes, D.; Meira, W. The evolution of bashlite and mirai iot botnets. In Proceedings of the 2018 IEEE Symposium on Computers and Communications (ISCC), Natal, Brazil, 25–28 June 2018; pp. 00813–00818. [Google Scholar]

[13] Cisco Annual Internet Report (2018–2023) White Paper. 2020. Available online: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html (accessed on 30 June 2021).

[14] Vasilomanolakis, E.; Karuppayah, S.; Mühlhäuser, M.; Fischer, M. Taxonomy and survey of collaborative intrusion detection. Acm Comput. Surv. 2015, 47, 1–33. [Google Scholar] [CrossRef]

[15] Summerville, D.H.; Zach, K.M.; Chen, Y. Ultra-lightweight deep packet anomaly detection for Internet of Things devices. In Proceedings of the 2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC), Nanjing, China, 14–16 December 2015; pp. 1–8. [Google Scholar]

[16] Midi, D.; Rullo, A.; Mudgerikar, A.; Bertino, E. Kalis—A system for knowledge-driven adaptable intrusion detection for the Internet of Things. In Proceedings of the 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Atlanta, GA, USA, 5–8 June 2017; pp. 656–666. [Google Scholar]

[17] Alothman, Z.; Alkasassbeh, M.; Al-Haj Baddar, S. An efficient approach to detect IoT botnet attacks using machine learning. J. High Speed Netw. 2020, 26, 241–254. [Google Scholar] [CrossRef]

[18] Aburomman, A.A.; Reaz, M.B.I. Review of IDS development methods in machine learning. Int. J. Electr. Comput. Eng. 2016, 6, 2432–2436. [Google Scholar]

[19] Bijalwan, A. Botnet forensic analysis using machine learning. Secur. Commun. Netw. 2020, 2020, 9302318. [Google

Scholar] [CrossRef]

[20] Shafiq, M.; Tian, Z.; Sun, Y.; Du, X.; Guizani, M. Selection of effective machine learning algorithm and Bot-IoT attacks traffic identification for internet of things in smart city. Future Gener. Comput. Syst. 2020, 107, 433–442. [Google Scholar] [CrossRef]

[21] Soe, Y.N.; Feng, Y.; Santosa, P.I.; Hartanto, R.; Sakurai, K. Machine learning-based IoT-botnet attack detection with sequential architecture. Sensors 2020, 20, 4372. [Google Scholar] [CrossRef]

[22] Diro, A.A.; Chilamkurti, N. Distributed attack detection scheme using deep learning approach for Internet of Things. Future Gener. Comput. Syst. 2018, 82, 761–768. [Google Scholar] [CrossRef]

[23] Ahmad, I.; Basheri, M.; Iqbal, M.J.; Rahim, A. Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection. IEEE Access 2018, 6, 33789–33795. [Google Scholar] [CrossRef]

[24] Deng, L.; Li, D.; Yao, X.; Cox, D.; Wang, H. Mobile network intrusion detection for IoT system based on transfer learning algorithm. Clust. Comput. 2019, 22, 9889–9904. [Google Scholar] [CrossRef]

[25] Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An ensemble of autoencoders for online network intrusion detection. arXiv 2018, arXiv:1802.09089. [Google Scholar]

[26] Ustebay, S.; Turgut, Z.; Aydin, M.A. Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In Proceedings of the 2018 International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT), Ankara, Turkey, 3–4 December 2018; pp. 71–76. [Google Scholar]

[27] Da Costa, K.A.; Papa, J.P.; Lisboa, C.O.; Munoz, R.; de Albuquerque, V.H.C. Internet of Things: A survey on machine learning-based intrusion detection approaches. Comput. Netw. 2019, 151, 147–157. [Google Scholar] [CrossRef]

[28] Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-baiot—Network-based detection of iot botnet attacks using deep autoencoders. IEEE Pervasive Comput. 2018, 17, 12–22. [Google Scholar] [CrossRef] [Green Version]

[29] Xie, H.; Wei, S.; Zhang, L.; Ng, B.; Pan, S. Using feature selection techniques to determine best feature subset in prediction of window behaviour. In Proceedings of the 10th Windsor Conference: Rethinking Comfort, Windsor, UK, 12–13 April 2018; pp. 315–328. [Google Scholar]

[30] Breiman, L. Random forests. Mach. Learn. 2001, 45, 5–32. [Google Scholar] [CrossRef] [Green Version]

[31] Genuer, R.; Poggi, J.M.; Tuleau-Malot, C. Variable selection using random forests. Pattern Recognit. Lett. 2010, 31, 2225–2236. [Google Scholar] [CrossRef] [Green Version]

[32] Parsa, A.B.; Movahedi, A.; Taghipour, H.; Derrible, S.; Mohammadian, A.K. Toward safer highways, application of XGBoost and SHAP for real-time accident detection and feature analysis. Accid. Anal. Prev. 2020, 136, 105405. [Google Scholar] [CrossRef] [PubMed]

[33] Friedman, J.H. Greedy function approximation: A gradient boosting machine. Ann. Stat. 2001, 29, 1189–1232. [Google Scholar] [CrossRef]

[34] Awal, M.A.; Masud, M.; Hossain, M.S.; Bulbul, A.A.M.; Mahmud, S.H.; Bairagi, A.K. A novel bayesian optimization-based machine learning framework for COVID-19 detection from inpatient facility data. IEEE Access 2021, 9, 10263–10281. [Google Scholar] [CrossRef]

[35] Htwe, C.S.; Thant, Y.M.; Thwin, M.M.S. Botnets Attack Detection Using Machine Learning Approach for IoT Environment. J. Phys. Conf. Ser. 2020, 1646, 012101. [Google Scholar] [CrossRef]

[36] Alharbi, A.; Alosaimi, W.; Alyami, H.; Rauf, H.T.; Damaševičius, R. Botnet Attack Detection Using Local Global Best Bat Algorithm for Industrial Internet of Things. Electronics 2021, 10, 1341. [Google Scholar] [CrossRef]

[37] Mason, S.J.; Graham, N.E. Areas beneath the relative operating characteristics (ROC) and relative operating levels (ROL) curves: Statistical significance and interpretation. Q. J. R. Meteorol. Soc. A J. Atmos. Sci. Appl. Meteorol. Phys. Oceanogr. 2002, 128, 2145–2166. [Google Scholar] [CrossRef]

[38] Abbas, A.; Khan, M.A.; Latif, S.; Ajaz, M.; Shah, A.A.; Ahmad, J. A New Ensemble-Based Intrusion Detection System for Internet of Things. Arab. J. Sci. Eng. 2021, 1–15. [Google Scholar] [CrossRef]

[39] Goeschel, K. Reducing false positives in intrusion detection systems using data-mining techniques utilizing support vector machines, decision trees, and naive Bayes for off-line analysis. In Proceedings of the SoutheastCon 2016, Norfolk, VA, USA, 30 March–3 April 2016; pp. 1–6. [Google Scholar] [CrossRef]

[40] Hezam, A.A.; Mostafa, S.A.; Ramli, A.A.; Mahdin, H.; Khalaf, B.A. Deep Learning Approach for Detecting Botnet Attacks in IoT Environment of Multiple and Heterogeneous Sensors. In Proceedings of the International Conference on Advances in Cyber Security, Penang, Malaysia, 24–25 August 2021; Springer: Singapore, 2021; pp. 317–328. [Google Scholar]

[41] Khoa, T.V.; Saputra, Y.M.; Hoang, D.T.; Trung, N.L.; Nguyen, D.; Ha, N.V.; Dutkiewicz, E. Collaborative Learning Model for Cyberattack Detection Systems in IoT Industry 4.0. In Proceedings of the 2020 IEEE Wireless Communications and Networking Conference (WCNC), Seoul, Korea, 25–28 May 2020; pp. 1–6. [Google Scholar] [CrossRef]