ECBoA-OFS: An Ensemble Classification Model for Botnet Attacks based on Optimal Feature Selection using CPR in IoT

[1]Chandana Swathi G, [2]Kishor Kumar G and [3]Siva Kumar A P

[1,3]Department of Computer Science and Engineering, Jawaharlal Nehru Technological University Anantapur College of Engineering (Autonomous), Ananthapuramu, Andhra Pradesh, India.

[2]Department of Computer Science and Engineering, Rajeev Gandhi Memorial College of Engineering and Technology, Nanadyal, Andhra Pradesh, India.

[1]chandanaswathisura@gmail.com, [2]kishorgulla@yahoo.co.in, [3]sivakumar.cse@jntua.ac.in

Correspondence should be addressed to Chandana Swathi G : chandanaswathisura@gmail.com

Article Info

Journal of Machine and Computing (http://anapub.co.ke/journals/jmc/jmc.html)

Doi : https://doi.org/10.53759/7669/jmc202404081

Received 10 March 2024; Revised from 22 May 2024; Accepted 16 July 2024.

Available online 05 October 2024.

Abstract – The rapid growth of the Internet of Things (IoT) has indeed introduced new security challenges, and the proliferation of compromised IoT devices has become a significant concern. Botnet attacks, where multiple corrupted devices are managed by a particular object, have become a widespread threat in IoT environments. These are used for a variety of malicious activities, including distributed DDoS attacks, data breaches, and malware distribution. However, detecting IoT botnets poses several challenges due to the resource constraints inherent in many IoT devices. The limitations in computation, storage, and communication capabilities make it challenging to deploy complex ML and deep learning models directly on these devices. This paper proposes an ensemble classification model ECBoA-OFS (Ensemble Classification for Botnet Attack Prediction using Optimal Feature Selection). It focuses on enhancing the accuracy of botnet attack prediction through the integration of ensemble methods and optimal feature selection. It describes a method for optimal feature selection in the context of analyzing the behavior of BoA and malicious traffic flow features in a network using Central Pivot Ranges (CPR). Feature selection is an important step in machine learning and data analysis because it supports to identification of the most important features for a given problem, thereby improving model performance and interpretation. The extracted features are used for model training and ensemble classification for prediction. To evaluate ECBoA-OFS, the N-BaIoT-2021 dataset consisting of regular IoT network traffic and BoA traffic records of corrupted IoT devices is utilized, considering detection precision, sensitivity, specificity, accuracy, and F1-score. Although all ensemble classifier models achieved better detection accuracy through optimal feature selection, the proposed ECBA-OFS shows better results compared to other ensemble classifier results.

Keywords – Internet of Things (IoT), Botnet Attacks, Feature Selection, Central Pivot Range, Ensemble Classification.

I. INTRODUCTION

The IoT (Internet of Things) refers to a network of physical devices, automobiles, buildings, and other objects surrounded by sensors, software, and network connectivity. These devices collect and share information, creating an ecosystem that communicates with each other and with centralized systems [1]. The past decade has seen a remarkable rise in the level of interconnectivity, giving rise to the IoT. The IoT signifies a standard move in the manner devices, machines, and services communicate and interact with each other [2]. This interconnected network enables seamless communication and data exchange, fostering a more efficient and integrated approach to various aspects of daily life, industry, and technology. The IoT indeed poses significant security challenges due to its widespread integration into various facets of daily life. The increasing prevalence and complexity of anomalies and security breaches on IoT devices pose a serious threat to the overall security of the IoT ecosystem.

The incorporation of Machine Learning (ML) with IoT services has indeed paved the way for innovative applications across diverse domains. To ensure the effectiveness of these applications, particularly in areas like security, surveillance, healthcare, transportation, control, and object monitoring, the precision and accuracy of ML models play a crucial role [3], [4]. It describes a strategy commonly used in cybersecurity called anomaly detection or behavioral analysis, where

ML is employed to model legitimate user behavior and identify deviations from this norm that may indicate potential security threats. So, designing IoT environments with security in mind is crucial to mitigate potential security threats.

A Botnet Attack on IoT devices, often referred to as an IoT Botnet Attack (BoA), is a kind of cyber threat where a network of compromised IoT devices is controlled remotely by attackers. In this scenario, the attackers typically exploit vulnerabilities in the security of IoT devices, allowing them to gain unauthorized access and control over these devices. One of the prominent threats associated with IoT botnets is their potential to launch DDoS attacks. This can result in service disruptions, financial losses, and damage to the targeted organization's reputation [5]. Mirai is a notorious malware that targets IoT devices, particularly those with weak security measures. It was responsible for several high-profile DDoS attacks, disrupting major websites and online services in 2016. The malware scans the internet for vulnerable IoT devices that still have default usernames and passwords, exploiting them to create a large network of suppressed nodes, identified as a botnet. Mirai's impact highlighted the security risks associated with poorly protected IoT devices, as many of these devices lacked basic security measures and were easily exploited. It also emphasizes the need for increased awareness and vigilance among device manufacturers, users, and the broader cybersecurity community to mitigate the risks posed by such malware threats.

Predicting and preventing IoT botnet attacks pose significant challenges due to the unique characteristics of IoT devices. The diversity of IoT Devices, heterogeneous configurations, restricted properties, and active environment of IoT make it highly challenging in anomaly detection and feature selections. To mitigate these limitations, researchers and experts are exploring innovative approaches such as ML models tailored for IoT environments, lightweight encryption techniques, behavior-based anomaly detection, and collaborative defense mechanisms that leverage information sharing among devices in the network. Additionally, ongoing efforts focus on developing standards and best practices for securing IoT devices and networks [6].

Utilizing ML techniques is favorable for IoT botnet prediction to solve the security challenges of IoT environments [7], [8]. In the context of IoT botnet prediction, ML can be used to detect unusual or malicious activities by learning from historical data. Traditional ML techniques, while effective in many cases, can struggle to adapt to and detect innovative or previously unseen threats. It faces limitations in handling advanced and evolving cyber-attacks due to limited training data, static feature representation, overfitting, and lack of context awareness. It's significant to observe that the area of cybersecurity is active, and researchers continually explore new methods to improve threat detection. A combination of traditional and more advanced ML techniques, along with a strong emphasis on real-time monitoring and adaptation is often recommended to enhance overall security posture. The passage underscores the evolving nature of cyber threats and the inadequacy of conventional ML models in addressing these challenges. The proposed solution involves adopting a more sophisticated approach through ensemble learning and classification to enhance the accuracy and adaptability of IoT attack detection systems [9].

Ensemble Learning (EL) involves combining multiple models to enhance overall performance. In this context, it suggests that a combination of models may be more effective in capturing the diversity of IoT attack patterns and improving detection capabilities. Over-fitting occurs when a model learns the training data very well and picks up noise or extraneous patterns that don't generalize well to invisible data. It can be a common problem, especially true when dealing with high-dimensional feature spaces as discussed in the context of traffic flow. This work intends to propose a solution through EL and classification methods to address the problem of over-fitting in traffic flow feature learning models.

This paper proposes a solution to address overfitting in the context of enhancing botnet classification and prediction accuracy. The proposed method employs Optimal Feature Selection (OFS) which is utilized in conjunction with an ensemble classifier model known as "ECBoA-OFS". It usually combines the predictions of several underlying classifiers to improve overall performance. It leverages the optimal feature selection method to enhance its classification and prediction accuracy. Feature selection is to choose the most relevant features and eliminate irrelevant or redundant ones, preventing overfitting and improving generalization to new data. By combining OFS with an ensemble classifier, the goal is to build a more robust and precise model for detecting botnet activity. The Central Pivot Ranges (CPR) technique is used to build OFS. Ensemble classification will improve overall prediction accuracy performance. The proposal will provide key benefits for using EL for anomaly detection, mainly from the perspective of detecting BoA. This will improve the robustness of the traffic flow features learning model and mitigate the risk of over-fitting.

We utilized the N-BaIoT2021 dataset [10] for detecting abnormal network behavior related to IoT botnets, specifically focusing on the Bashlite and Mirai vectors. This dataset is likely employed for developing and evaluating intrusion detection or anomaly detection systems that aim to identify compromised IoT devices within a network. It aims to contribute a system for improving the security of IoT devices by employing an EL strategy, optimized feature selection, and a classification model. The OFS uses CPR, and a focus on training efficiency to enhance IoT device security by accurately identifying botnet assaults through a detailed understanding of network traffic behavior.

The following structure of a paper is outlined; Section 2 provides related works. Section 3 presents methods and procedures for each major module. Section 4 presents evaluation results and performance analysis. Section 5 concludes the presented work.

II. RELATED WORKS

IoT is an interconnected system of various devices, sensors, networks, and applications. It has several common security issues including insecure device configurations, lack of firmware updates, insufficient encryption, and inadequate access controls [11]. Addressing these challenges needs a holistic process that integrates technical solutions, industry standards, and user awareness. The increasing prevalence of anomalies and security breaches in IoT devices is a significant concern. Additionally, collaboration among stakeholders, industry-wide standards, and regulatory frameworks play a vital role in enhancing the complete security posture of the IoT ecosystem. This capability to seamlessly integrate the physical world with digital systems can lead to innovative and advanced information services that benefit individuals, businesses, and society at large. However, as the IoT infrastructure framework becomes more complex and interconnected, it also introduces challenges and vulnerabilities that are necessary to be sensibly solved for each layer as shown in **Fig 1**.

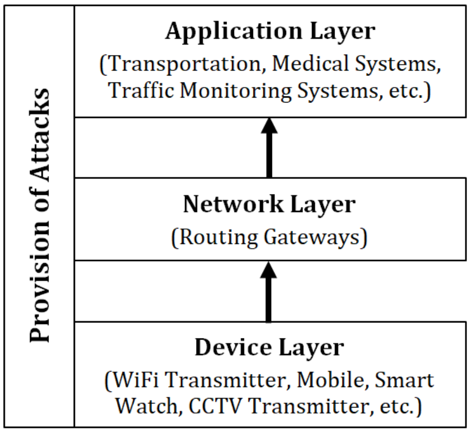

Provision of Attacks
Application Layer
(Transportation, Medical Systems, Traffic Monitoring Systems, etc.)
Network Layer
(Routing Gateways)
Device Layer
(WiFi Transmitter, Mobile, Smart Watch, CCTV Transmitter, etc.)

Fig 1. Provision of Attacks in IoT.

The intersection of IoT and ML has led to the development of numerous applications across diverse domains. While these innovations offer substantial benefits, they also introduce security challenges that need to be addressed to play a significant role in various aspects of communal life, prioritizing security is paramount to ensure a safe and trustworthy ecosystem. While ML offers significant advantages, it's essential to have a holistic approach to cybersecurity, including proper planning, implementation of best practices, and regular updates in enhancing security strategies for safeguarding IoT devices [12].

Feature Selection

Feature selection is a context of developing a predictive model, specifically for classification tasks like IoT detection from network traffic data. It aims to choose the most relevant and significant features from the original set of features that contribute the most to the predictive performance of the model. Dimensionality Reduction can reduce the number of data dimensions by selecting only the most relevant features. It overcomes the computational costs associated with modeling high-dimensional data and can improve the overall performance of the model. Therefore, by focusing on more useful features, the model can improve the detention of the underlying data and increase prediction accuracy. This is particularly crucial in classification tasks where accurately identifying patterns is essential. In the specific case of network traffic data for IoT detection, irrelevant or redundant features may not contribute meaningfully to distinguishing IoT devices from other network activities. Removing such features through feature selection can enhance the model's ability to identify patterns associated with IoT devices. It involves selecting the most relevant and informative features from a data set, which can improve model performance, decrease overfitting, and improve interpretability.

Nomm and Bahsi [13] presented a work on IoT botnet detection specifically mentioning their emphasis on feature selection using an unsupervised model. It proposes training a single model for all IoT devices instead of having dedicated models for each device, to achieve resource optimization. The proposed solution involves multiple methods of feature selection, including Hopkins statistics-based feature selection, entropy-based, and variance-based methods. It conducted sampling on the original unbalanced dataset to obtain a balanced dataset for analysis. SVM performed well on the unbalanced dataset, showing notable accuracy, in comparison to isolation forest with entropy-based feature selection, it outperformed other combinations of feature selection methods and models.

In [14] author describes a process related to dimensionality reduction using autoencoder models. Using an autoencoder for dimensionality reduction allows the extraction of meaningful information from the input data in a lower-dimensional space, facilitating tasks such as feature learning, data visualization, and noise reduction. Palmieri et al [15] propose an anomaly network detection using independent component analysis (ICA) in a distributed approach. Implementing ICA in the context of network anomaly detection can potentially enhance the ability to identify unusual patterns or deviations from normal behavior in the network. By separating independent components, this approach can provide a better empathy of the fundamental structure of network data, contributing to the overall effectiveness of anomaly detection systems.

CPR: Central Pivot Range

Central Pivot Range (CPR) [16] describes a method related to financial markets and technical analysis, specifically the use of pivot points. Pivot points are commonly used in trading to identify potential support and resistance levels for a given asset. These points are calculated based on the high, low, and close (or mean) prices of the previous period. It is emphasized, and it likely plays a crucial role in determining potential support and resistance levels. The CPR might refer to a range around the Central Pivot Point that traders use to gauge the significance of price movements. Pivot points can be calculated using various methods, and traders often use them to make decisions about entering or exiting trades. They are considered a technical analysis tool and are used to identify key levels that might influence the price movement of an asset.

CPR is a statistical calculation used in technical analysis, derived from the previous high, low, and close values. The three key levels in the CPR are the Central Pivot Point (CPP), Top Central Level (TCL), and Bottom Central Level (BCL). These are calculated using the Eq. (1), Eq. (2), and Eq. (3),

$$CPP = \frac{High\_Val + Low\_Val + Close\_Val}{3} \tag{1}$$

$$TCL = CPP + (High\_Val - Low\_Val) \tag{2}$$

$$BCL = CPP - (High\_Val - Low\_Val) \tag{3}$$

The CPR is a technical calculation from the make-spans or volumes of previous data observations. It can help traders identify potential support and resistance levels for the upcoming data. It is a sequence containing TCL, CPP, and BCL in that order. However, variations in the CPR range develop based on the volume and operating range of the indicator observed in the past.

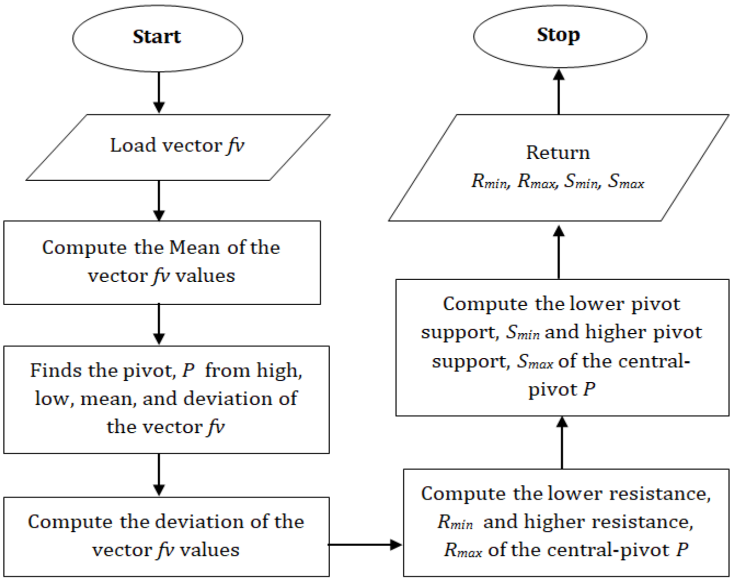

Start
Load vector *fv*
Compute the Mean of the vector *fv* values
Finds the pivot, *P* from high, low, mean, and deviation of the vector *fv*
Compute the deviation of the vector *fv* values
Compute the lower resistance, R_{min} and higher resistance, R_{max} of the central-pivot *P*
Compute the lower pivot support, S_{min} and higher pivot support, S_{max} of the central-pivot *P*
Return R_{min}, R_{max}, S_{min}, S_{max}
Stop

Fig 2. Data Flow Diagram for *pivotRanges( )* Function.

To perform optimal feature selection (OFS) using CPR, we treat the set of found values for each indicator as a vector, *fv,* and call the method *pivotRanges (fv)*, which receives the vector *fv* as a parameter and calculates two support (*S*) values and two resistance (*R*) values, which are used considered for feature selection. The data flow of the pivot calculation is shown in **Fig 2**. Based on the calculated resistance and support values, optimal feature selection is performed to obtain higher support with lower resistance features.

Ensemble Learning and Classification

Ensemble Learning (EL) is an influential technique in ML that combines predictions from different models to create a more reliable and accurate prediction model. They are generally less sensitive to noise and outliers in the data. Outliers can affect individual models disproportionately, but their impact is often reduced when combined with data from other models. This makes the ensemble more robust in the presence of noisy or outlying data points. The EL system around IDS refers to the process of merging several separate models to build a more robust, accurate, and reliable model for detecting and classifying network intrusions. This approach is widely recognized in the field for its ability to address the limitations of individual models while improving the overall performance of high-dimensional and dynamic data.

In the context of botnet attack detection on IoT devices, using an ensemble approach can enhance accuracy and resilience. It can achieve high accuracy, a low false positive rate, and resilience against adversarial attacks, surpassing the limitations of earlier methods in the field.

Ibrahim et al. [17] proposed an ML-based approach using a multilayer perceptron for botnet detection. The key focus of their framework is to overcome the limitations of traditional signature-based analysis, specifically in detecting unseen botnets that can evade such methods. The framework introduced is comprised of two main modules: filtering and classification. These modules utilize ML algorithms to accomplish the task of botnet detection. The filtering module likely preprocesses the data or extracts relevant features to improve the outcome of the subsequent classification module. The classification module, based on a multilayer perceptron, is designed to identify the attack command of the botnet and subsequently take control of the server. This suggests that the system not only detects the presence of a botnet but also aims to mitigate its impact by understanding and responding to the attack commands. It shows an enhanced accuracy of up to 92% and a low false-negative value of 1.5% suggesting the effectiveness of the framework in detecting botnet activities.

Alkahtani et al. [18] focus on detecting BoA in IoT applications utilizing a hybrid learning approach based on Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) for improved accuracy in detecting botnet attacks. It effectively classifies both benign and malicious patterns evaluating the N-BaIoT dataset. It specifically focuses on predicting the BoA from doorbell devices and thermostat devices. The experimental results indicate the efficiency of the CNN-LSTM model in predicting BoA with accuracy rates for doorbell devices reported as 90.88% and 88.61%, while thermostat devices achieve an accuracy rate of 88.53%.

Leevy et al. [19] focused on evaluating ensemble feature selection techniques (FSTs) in the context of classification performance for specific attack instances. It utilized a combination of EL (RF, XGBoost, Light GBM, CatBoost), non-EL (DT, LR, NB), and MLP (multi-layer perceptron). It suggests that ensemble FSTs may not directly impact classification performance but they prove beneficial due to other factors. Specifically, the ensemble FSTs were observed to offer advantages in terms of feature reduction, which can ease the computational burden. The combination of multiple FSTs in an ensemble, comprising both ensemble and non-ensemble learners, does not necessarily enhance classification performance directly. However, the benefits lie in the reduction of features, leading to a lighter computational load and improved data visualization, which can be valuable for understanding the dataset and gaining insights.

Al-Haija et al. [20] present an EL model designed for detecting BoA in IoT networks known as “ELBA-IoT”. The model focuses on behavioral characteristics specific to IoT networks and uses EL to recognize anomalous network traffic originating from negotiated IoT devices. It also evaluates the performance of three different ML techniques falling under decision tree methods. The experimentation utilizes the N-BaIoT2021 dataset, and according to the results analysis, there is an observed improvement in detection accuracy, reaching approximately 99.6%.

Rezaei [21] proposed a technique for the prediction of BoA in IoT utilizing the EL technique termed ELT-DB. It aimed to improve accuracy in botnet detection on IoT devices while minimizing the number of features required. It combines supervised, unsupervised, and regression approaches to learn botnet features to optimize accuracy and reduce the number of features needed for detection. The strengths of the ML technique were leveraged in forming the ensemble, but the union of selected features might lead to overlooking redundancy and irrelevance. This could potentially result in a larger feature set, which may have implications for efficiency and interpretability. This work result achieves a high accuracy rate of around 99% for botnet detection on IoT devices through the use of several tuning learning data features. It should be noted that although high accuracy is a positive result, the possibility of a larger set of features due to association work must be considered.

Shafiq et al. [22] propose a new framework model for malicious bot-IoT traffic detection employing a feature selection metric based on CorrAUC in combination with TOPSIS and Shannon entropy for evaluating the chosen feature sets. The CorrAUC effectively filters features and selects output features for the selected ML algorithm by measuring the area under curvature (AUC). Analysis of results showed that the proposed method was effective and the average accuracy was more than 96%. The method relies on random entropy, which makes probabilistic decisions to determine feature accuracy. The use of the AUROC curve adapts to select an ML algorithm, emphasizing the need for highly accurate features and the models verified on datasets with defined features, demonstrating high sensitivity and specificity. However, decision accuracy in real-time scenarios suggests potential challenges in practical applications.

The IoT-Botnet database focuses on detecting attacks on IoT networks by selecting EL-based features and developing classifiers to detect malicious traffic in IoT networks [23]. Studying the effect of ensemble classifiers on the performance of different attack classes highlights the importance of optimal feature selection to improve efficiency compared to individual selection methods. In the next section, we discuss the feature selection process and its effect on prediction, an important aspect of ML, especially in intrusion detection and security in IoT networks. It includes the discussion of the optimal feature selection process along with the Ensemble Classifiers and their effect on model performance with consideration of various attack classes.

III. PROPOSED METHODOLOGY

The proposed ECBoA-OFS is designed for processing and classifying traffic in an IoT network, specifically focusing on botnets for normal and malicious behaviors. The OFS Module is responsible for selecting the most relevant features from the input dataset (N-BaIoT 2021 dataset). Feature selection reduces dimensionality and improves the efficiency and effectiveness of the model. It includes using EL techniques to improve overall performance and generalization capabilities through a sequential process of processing and predicting selected features.

Optimal Features Selection (OFS)

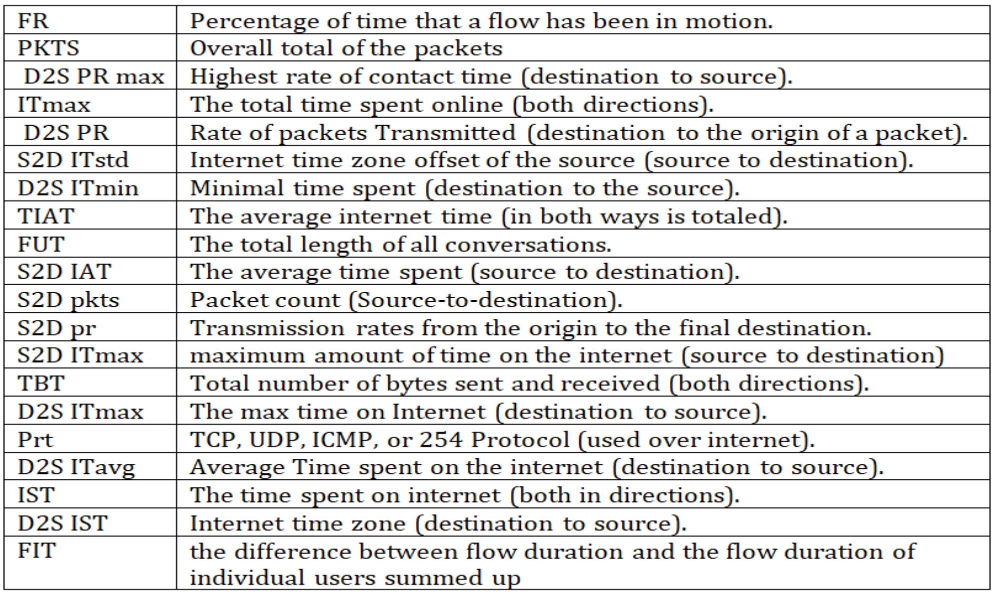
N-BaloT2021 dataset [10] is associated with a large number (*611,359 samples)* of features for training a classifier to detect anomalous network behavior, specifically related to IoT botnet attacks. It dataset contains both normal and botnet traffic samples, with the botnet traffic further categorized into Bashlite *(4737 samples)* and Mirai *(3000 samples)* attacks. It major concern is that having a large number of features for training the classifier might lead to process complexity, weak conditions for botnet detection, and potentially result in false alarms. In the general observation in supervised learning having more than 50 features may lead to overfitting, and it is important to find a balance between having enough features for accurate classification and avoiding underfitting. To address these concerns and optimize the classification process, we propose arranging the extracted features from the dataset in a two-dimensional matrix format, where each column represents the values obtained for each feature in the form of a malicious or benign class. This behavior is common in ML, where every record resembles a sample of data and every column resembles a feature. It's essential to carefully select relevant features and potentially employ feature selection techniques to reduce dimensionality and improve the efficiency of the classifier to overcome the challenges in botnet detection by optimizing feature selection.

Features Selection

The feature selection process for identifying optimal botnet features using a diversity assessment measure by using the Wilcoxon rank-sum test (WRST) [24] method. It works on the values of each feature, treating them as two vectors corresponding to the attack, and benign labels are represented as two vectors fv_a, fv_b.

The WRST which is a non-parametric test used to compare two independent samples with a two-sided test has been performed, indicating an interest in differences between populations without specifying a particular direction. When a one-tailed test is employed it will show a clear direction of interest, as True or false changes in one inhabitant relative to another. The selection of a one-tailed or two-tailed test relies on the exploration demand and the way of the anticipated results. The testing process involves combining observations of two data into a single data and noting to which sample each reflection belongs. It allows the comparison of distributions without making assumptions about the underlying population distribution. Then, the identified samples are sorted in ascending order from 1 to $n_1 + n_2$, and selects the top 20 features from the 204 identified features from the sample, as listed in **Table 1**.

Table 1. Features Selected using WSRT



FR	Percentage of time that a flow has been in motion.
PKTS	Overall total of the packets
D2S PR max	Highest rate of contact time (destination to source).
ITmax	The total time spent online (both directions).
D2S PR	Rate of packets Transmitted (destination to the origin of a packet).
S2D ITstd	Internet time zone offset of the source (source to destination).
D2S ITmin	Minimal time spent (destination to the source).
TIAT	The average internet time (in both ways is totaled).
FUT	The total length of all conversations.
S2D IAT	The average time spent (source to destination).
S2D pkts	Packet count (Source-to-destination).
S2D pr	Transmission rates from the origin to the final destination.
S2D ITmax	maximum amount of time on the internet (source to destination)
TBT	Total number of bytes sent and received (both directions).
D2S ITmax	The max time on Internet (destination to source).
Prt	TCP, UDP, ICMP, or 254 Protocol (used over internet).
D2S ITavg	Average Time spent on the internet (destination to source).
IST	The time spent on internet (both in directions).
D2S IST	Internet time zone (destination to source).
FIT	the difference between flow duration and the flow duration of individual users summed up

Optimization of Features

A feature selection process using the correlation coefficient value (CCV) between botnet features and traffic flow features is described here. This approach is commonly used in ML to recognize the highest relevant features that add positively to the model's results, particularly in the environment of identifying and dealing with botnet activities in network traffic. The selection of features depends on their association with the destination response, specifically using the Pearson correlation coefficient (PCC) [25]. Features that are positively correlated with the target are selected for inclusion in the model. Positive correlation specifies that as the significance of a feature increases, the target response value also tends to increase and a negative impact on the model's performance is avoided. A negative correlation suggests that as the value of a feature increases, the target variable's value tends to decrease. PCC calculates the strength and way of the linear association among every feature and the destination variable using Eq. (4). So, using the CVV for feature selection can help reduce processing overhead and ensure that only relevant features are measured.

$$\rho(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \cdot \sigma_Y} \tag{4}$$

where, $$\operatorname{cov}(X,Y) = \frac{1}{N}\sum_{i=1}^{N}(X - mean(x))(Y - mean(y))$$

Here, the Pearson correlation coefficient $\rho(X,Y)$ is calculated *cov(X, Y)* (covariance) divided by $\sigma_X \cdot \sigma_Y$ (standard deviations), The range of value lies between -1 and 1. A positive result specifies a positive association, a negative result specifies a negative association and a value of 0 specifies no linear association. The association between traffic and botnet features is used to select standard botnet features among the selected best botnet features to train the classifier, as shown in **Table 2**.

The process of estimating the scope and impact of botnet attacks using certain optimal features, as well as calculating confidence scores for each botnet feature in a given unlabelled record to do botnet detection. For each unlabelled traffic flow record, extract and analyze the relevant features to estimate the scope of botnet attacks based on traffic flow features. Based on the values derived from both traffic flow features and botnet features, categorize the scope of botnet attacks into four classes: "severe", "moderate", "mild" and "benign". **Table 3** defines the conditions for each class based on the severity of the attack as reflected in the features.

Table 2. Optimal Features Selected Between Traffic Flow and Botnet Features

FR	Percentage of time that a flow has been in motion.
ITmax	The total time spent online (both directions).
S2D ITstd	Internet time zone offset of the source (source to destination).
S2D IAT	The average time spent (source to destination).
S2D pkts	Packet count (Source-to-destination).
S2D ITmax	maximum amount of time on the internet (source to destination)
TBT	Total number of bytes sent and received (both directions).
FIT	the difference between flow duration and the flow duration of individual users summed up

Table 3. Class Label and Their Impact Condition

Class Label	Condition
Sever	Botnet impact is sever, if Count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards botnet label are 80% and above and Count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards benign label are 20% and less
Moderate	botnet impact is moderate, if count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards botnet label are 60% and above and the count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards benign is 40% and less
Mild	botnet impact is mild, if count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards botnet label are 40% and above and the count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards benign is 60% and less
Benign	botnet impact is mild, if count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards botnet label are 20% and less and the count of features $(\lvert TFF \rvert + \lvert BF \rvert)$ towards benign are 80% and above

ECBoA-OFS Classification

A classification methodology called "ECBoA-OFS" is proposed for botnet attack detection on a benchmark dataset that contains examples of botnet attacks. The benchmark dataset is divided into two portions through a dataset-splitting process. One portion is used for training the classification model and the other portion is used for testing the model. The ensemble approach of the methodology combines three different classification algorithms, RF (Random Forest, AB (AdaBoost), and KNN (K-Nearest Neighbors). The combination of these algorithms is referred to as the "RAK technique". This technique utilizes fully optimized features, specifically selected from traffic flow features and botnet features to achieve efficient botnet attack detection results. **Fig 3** illustrates the proposed ECBoA-OFS Classification methodology. It likely provides a graphical representation of the workflow or components of the proposed approach.

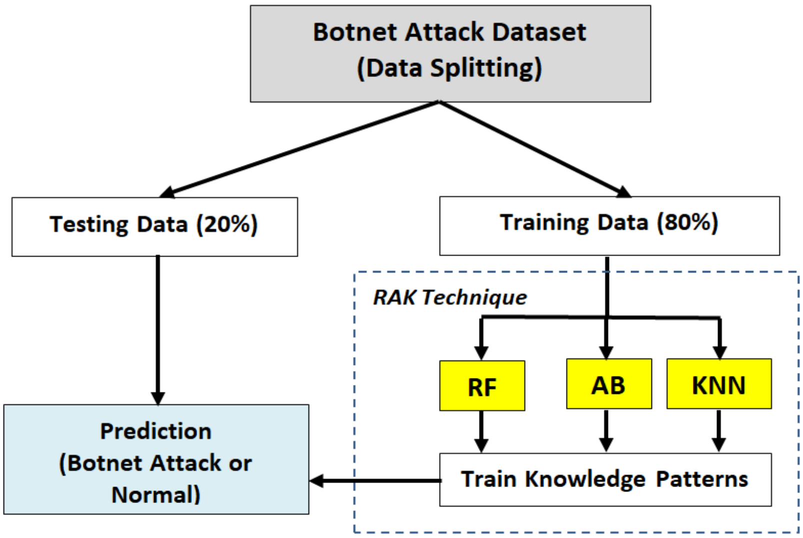


Fig 3. Methodology for Botnet Attack Detection using RAK Technique.

Data Splitting

Splitting data into training and test sets is a general process in ML to evaluate a model's performance and generalization ability. The 80:20 splitting ratio is chosen for the splitting, but the specific ratio can fluctuate depending on the size of the dataset and the kind of research problem. This helps identify biases that can occur when a model performs well on training data but fails to generalize to new observable data. Assessing the model on a distinct test set allows for determining whether it has learned specific outlines in the training data or can create accurate detections for unidentified cases.

Ensemble Learning Methods

RF (Random Forest)

RF is a recognized ML algorithm that leverages the concept of EL to combine multiple models to increase prediction accuracy [26, 27]. The algorithm generates several DTs, each trained on a subset of arbitrary data and features which helps in reducing overfitting. The results from distinct DTs are aggregated to create an ultimate detection. For classification tasks, this might involve a majority vote, while for regression tasks, it could be an average. It is known to be simpler, less prone to overfitting compared to individual DT, and capable of handling high-dimensional datasets with various features. Each DT in RF is prepared with different training data sets and arbitrary feature collections. These differences help improve the robustness and accuracy of the overall model. The assessment rules and result possibilities for individual DTs as well as the ensemble of the RF.

The RF consists of a set of n DT. Each DT is trained on a features subset (F_i) and training data (D_i.). The branches, B_i. j represents prospective results or predictions. Every node j in a DT i represents a selection or coincidental occurrence based on the training data D_i, j. Branches $B_i.j.k$ lead to sub-node l and have a possibility of happening p_i, j, k. The collection of branches $B_i.j.k$ can stand for potential outcomes or their probabilities. Conditional possibility or combined probability is used to describe the probability of each event taking into account legal rules and precedents. RF improves overall performance and flexibility by combining information from multiple decision trees. The final number has always been obtained by adding the numbers for all trees.

RF is an EL method that works by building multiple DTs during training and extracting the prediction method or average of the individual trees. Each tree in the set provides a prediction, and the summation process determines the final prediction or estimate. They are known for capturing complex relationships in data, reducing overfitting, and providing robust and accurate predictions. Due to their efficiency and flexibility, they are widely used in various ML tasks, including classification and regression.

AdaBoost (AB)

AdaBoost is a standard EL method implemented for both regression and classification tasks in ML. It stands for Adaptive Boosting, and it belongs to the family of boosting algorithms. The algorithm starts by assigning equal weights to all the training instances. A weak classifier (often a decision tree with limited depth) is trained on the data. It focuses on getting the correct classification for the instances that were misclassified in the previous rounds. The performance of the weak classifier is evaluated, and instances that were misclassified receive higher weights. The weight of each bad rater's decision is determined by its accuracy and its event values are updated according to true or false classification. Misclassified events receive more weight so that they can be given more importance in the next round. Repeat the previous 2 steps for a determined number of cycles or until the specified level of accuracy is reached. The final model is a mixture of weights for all dynamic classes, where the weights are determined by their performance in the training process. It is famous for its ability to improve the accuracy of weak learners and create reliable and accurate models. It tends to extremes and extremes. However, it is sensitive to noisy data and outliers.

The algorithm iteratively allocates weights to events in the training data, focusing more on instances that are misclassified by the current set of weak classifiers. Let $T= \{t_1, t_2, . . . , t_n \}$ represent the training dataset with n instances. Each instance t_i is accompanied by a class c_i. Let $W= \{w_1, w_2, . . . , w_T \}$ is a set of weak classifiers. Each weak classifier w_t takes an event t as input and outputs a binary detection $w(t) \in \{-1, 1\}$. Let $D_t = \{d_{t,1}, d_{t,2}, \ldots, d_{t,n} \}$ represent the weight sharing over the training data at repetition x. Each $d_{t,i}$ is the weight allocated to events i at repetition x. The weights are adjusted in all analyses based on the number of samples that were misclassified in the first analysis. It aims to improve the overall performance by giving more emphasis to instances that are challenging to classify, making it particularly effective in situations with noisy or complex data.

K-Neighbors classifier (KNN)

The KNN algorithm is a common learning algorithm used in ML classification tasks [28]. It involves finding the k-nearest neighbors of a new record and assigning a class to a new group based on the number of neighboring classes. The KNN algorithm is modest and effective and is often placed in the base model for assessment with more difficult algorithms. However, KNN has some limitations. This means that performance can deteriorate as the number of features or dimensions in data increases due to the curse of size. In addition, KNN can be computationally costly, particularly for big data sets, because it has to calculate the difference between a new and all current data points.

Let X be the feature space, x_i be a data point in X, and Y be the corresponding output variable (class label). The KNN algorithm expects the class label for a fresh data point x by considering the majority class among its k- k-nearest neighbors. The distance between two instances, $x_{i,}$ and x_j , in a dataset, can be calculated by implementing several distance factors such as "Euclidean Distance", "Manhattan Distance (L1 Norm)" and "Minkowski Distance"**.** The estimate for the new data point x is determined by a state vote. KNN is a non-parametric delayed learning algorithm utilized for both classification and detection tasks.

Ensemble models have proven to be highly effective in various domains, including network attack detection. The concept of EL contains merging several models to build a stronger and additional reliable predictive model than any individual model. This approach is particularly beneficial in the context of network security, where the accuracy and robustness of attack detection systems are of utmost importance. Their ability to leverage diverse models, reduce overfitting, improve accuracy, and provide robustness makes them a valuable approach in the ever-evolving landscape of cybersecurity.

IV. EXPERIMENT EVALUATION

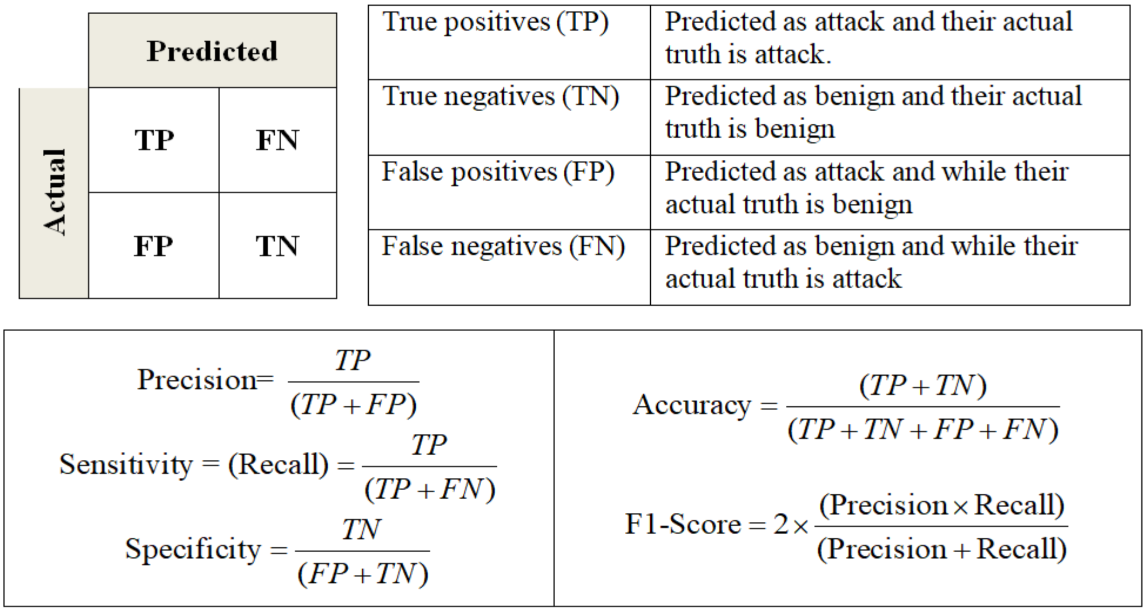
The N-BaIoT dataset [10] is chosen for experiment evaluation due to its relevance to the focus on IoT security. Since it is set from an IoT context, it aligns well with the research objectives. The dataset contains inserted malicious traffic related to BoA. This inclusion allows for a realistic evaluation of the particular ML algorithms in identifying and qualifying BoA, which is crucial for assessing their effectiveness. The presence of injected botnet attacks in the dataset creates a realistic scenario for evaluation. This characteristic simplifies the assessment of the proposed ECBoA-OFS results with state-of-the-art works, providing a standardized basis for evaluating their effectiveness.

Evaluation Measures

The evaluation for the ECBoA-OFS model is performed using standard evaluation measures [29] over k-fold datasets to obtain a comprehensive understanding of its performance across different impact class variants. The evaluation metrics —Precision, Sensitivity, Specificity, Accuracy, and F1-Score are utilized to assess the classification outcomes of a model [30]. The confusion matrix table summarizes the model's predictions as given in **Fig 4**.

Result Analysis

A comparative analysis with different ensemble models such as ELBA-IoT [20], ELT-DB [21], and CorrAUC [22] against the proposed model ECBoA-OFS is performed. The evaluation is based on 4-Folds Cross Validation, and the classes being considered are "Severe", "Moderate", "Mild", and "Benign". The 4-Folds Cross Validation divides the dataset into four subsets or folds. The model was developed for 4-folds and tested on the remaining data. This method is iterated four times, each time using a diverse test set. Evaluation criteria are then calculated based on these cross-validation results.



	Predicted	
Actual	TP	FN
	FP	TN

True positives (TP)	Predicted as attack and their actual truth is attack.
True negatives (TN)	Predicted as benign and their actual truth is benign
False positives (FP)	Predicted as attack and while their actual truth is benign
False negatives (FN)	Predicted as benign and while their actual truth is attack

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

$$\text{Sensitivity} = (\text{Recall}) = \frac{TP}{(TP+FN)}$$

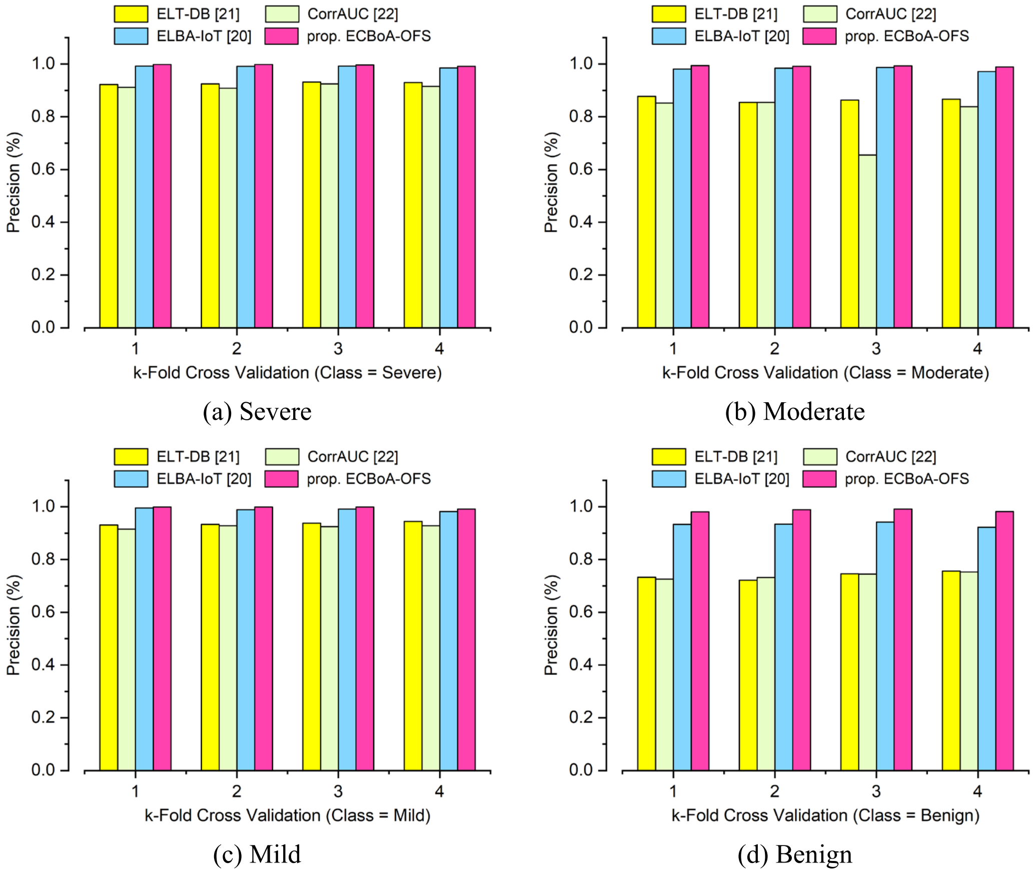
$$\text{Specificity} = \frac{TN}{(FP+TN)}$$

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

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

Fig 4. Evaluation Metrics and Measures.

Precision Comparison Analysis



(a) Severe

(b) Moderate

(c) Mild

(d) Benign

Fig 5. Precision Comparison Analysis for Different Attack Class.

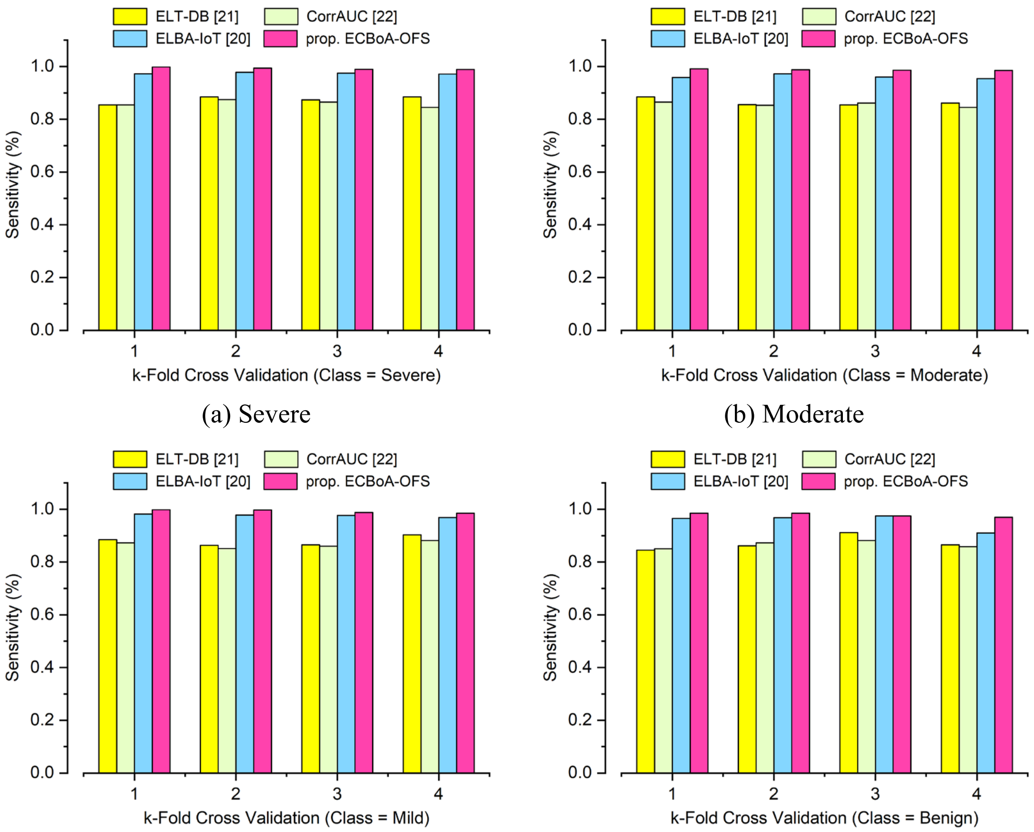
Fig 5 illustrates precision comparison analysis with four impact classes. Precision is the ratio of TP calculations to the total number of True detections identified by the model, and it assesses the accuracy of True detections. The comparison analysis shows ECBoA-OFS performs better compared to other cases across the four impact classes. Precision values are likely higher for ECBoA-OFS in each impact class. **Table 4** provides a comprehensive comparison of the average performance for each impact class. These metrics or values summarize the model's overall performance across the different impact classes.

Table 4. Average Precision Result Comparison

Class	**ELT-DB [21]**	**CorrAUC [22]**	**ELBA-IoT [20]**	**prop. ECBoA-OFS**
Severe	0.927	0.915	0.990	0.996
Moderate	0.866	0.800	0.981	0.992
Mild	0.924	0.989	0.997	0.924
Benign	0.739	0.739	0.933	0.986

The performance of precision metrics of the proposed method ECBoA-OFS has shown improvement in comparison to others, reaching a highest value of 0.996% with severe class prediction and a lowest value of 0.924% with mild class prediction. Generally, a higher precision value specifies a lower rate of FP, which significance the model predicts is more likely to be true.

Sensitivity Comparison Analysis



(a) Severe

(b) Moderate

(c) Mild

(d) Benign

Fig 6. Sensitivity Comparison Analysis for Different Attack Class.

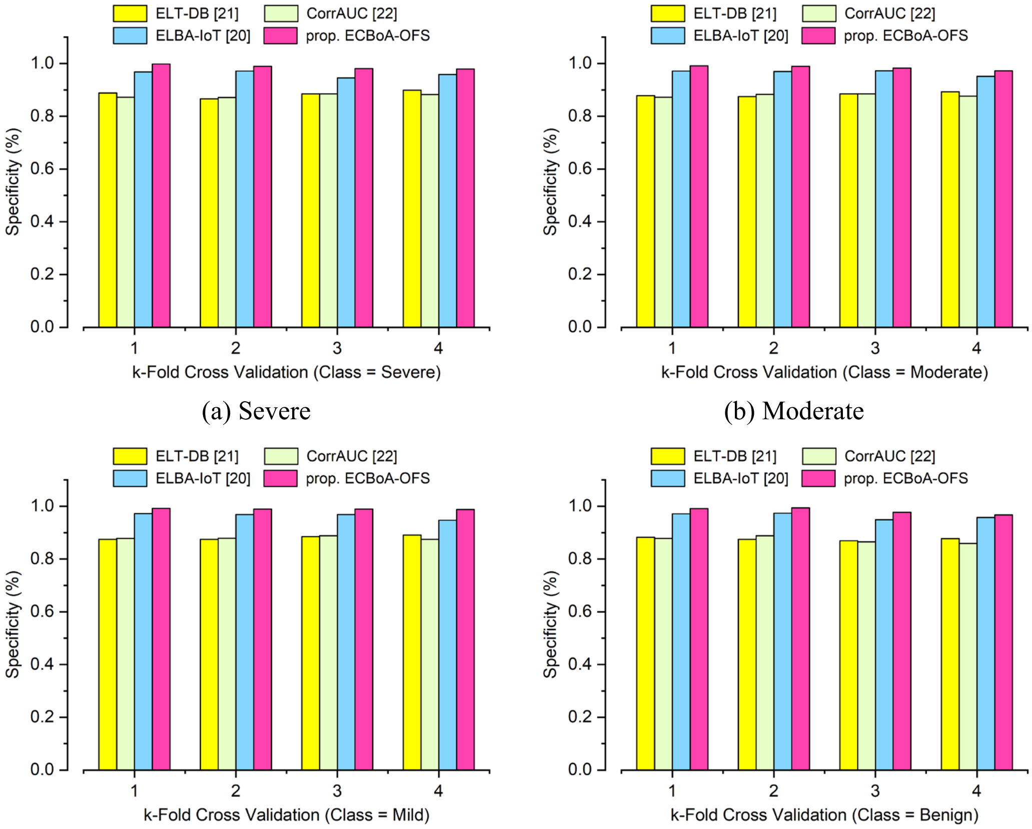
Fig 6 depicts a sensitivity comparison analysis with four impact classes. Sensitivity (recall) or TP rate, is a measure that assesses the model's capability to predict all True instances. It indicates how well the model identifies instances in each of the four impact classes. The proposed ECBoA-OFS demonstrates better results across all four impact classes compared to other cases. This suggests that ECBoA-OFS has a higher true positive rate or recall for positive instances in each class. **Table 5** will provide an average performance comparison for every class and present a further comprehensive view of the model's outcome through the diverse impact classes.

Table 5. Average Sensitivity Result Comparison

Class	**ELT-DB [21]**	**CorrAUC [22]**	**ELBA-IoT [20]**	**prop. ECBoA-OFS**
Severe	0.875	0.860	0.974	0.993
Moderate	0.865	0.856	0.961	0.988
Mild	0.879	0.866	0.976	0.992
Benign	0.871	0.866	0.955	0.979

Sensitivity computes the capability of a model to suitably recognize true instances concerning the traffic sample tested. The proposed ECBoA-OFS shows a high sensitivity score, which indicates the model's efficiency in predicting instances of severe predictions, even though it may result in a few false positives. A high sensitivity score with the highest 0.993% and lowest 0.979% suggests the model is capturing a significant portion of the positive instances for all class predictions. It describes "better efficiency in predicting negative results" which states the potential of the classifier performing well in identifying instances where there are no intrusions (benign). The lowest score of 0.979% for benign predictions shows the model is slightly less sensitive to benign instances but still maintains a reasonably high level of accuracy.

Specificity Comparison Analysis



(a) Severe

(b) Moderate

(c) Mild

(d) Benign

Fig 7. Specificity Comparison Analysis for Different Attack Class.

Fig 7 illustrates a specificity comparison analysis with four impact classes. Specificity is defined as the ratio of TN predictions to the total number of complete negative instances. It essentially computes the model's capability to correctly recognize negative instances. The results indicate that in all four impact classes, the proposed ECBoA-OFS demonstrates better specificity compared to other cases. A comparison analysis of the average performance for each class is given in **Table 6**.

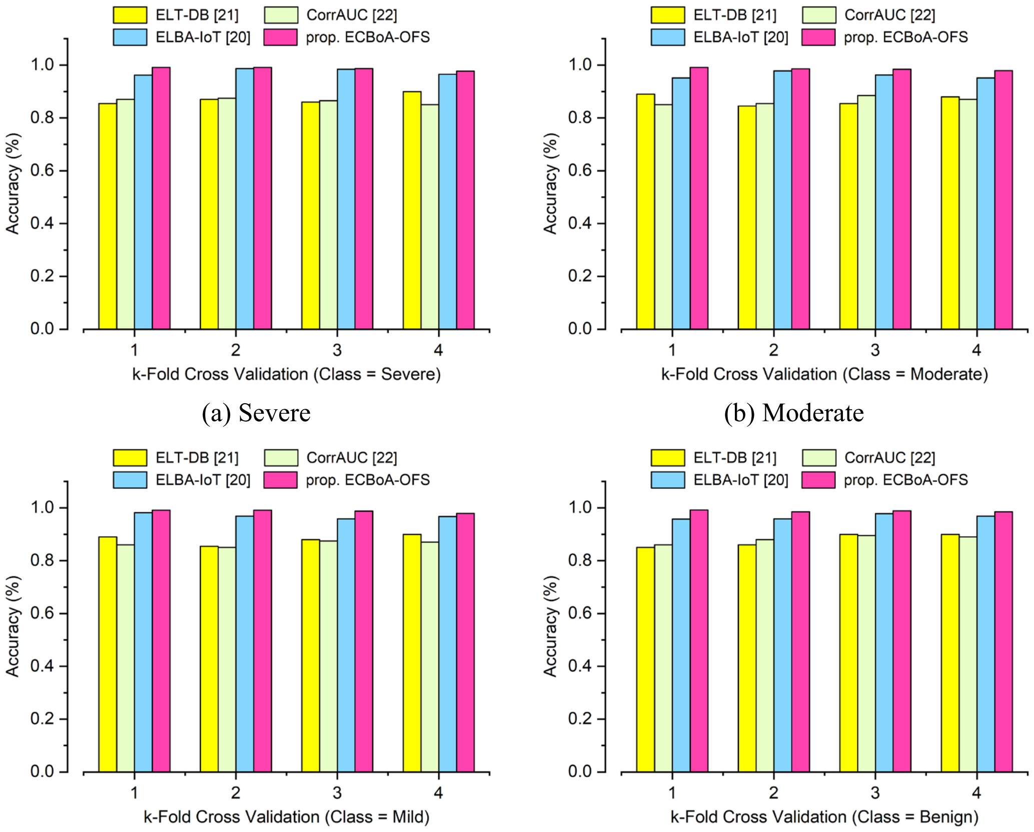
The specificity metrics and the performance of the proposed ECBoA-OFS for predicting negative results are given in **Table 6**. It measures the classifier's ability to properly detect TN out of complete negatives. The proposed ECBoA-OFS demonstrates better efficiency in predicting negative results and provides the highest specificity values of 99.0% for mild and the lowest specificity of 98.2% for benign. It suggests the effectiveness of correctly identifying negative results, and it is desirable to have high specificity for avoiding false positives (incorrectly predicting a positive result).

Table 6. Average Specificity Result Comparison

Class	ELT-DB [21]	CorrAUC [22]	ELBA-IoT [20]	prop. ECBoA-OFS
Severe	88.5	87.8	96.1	98.7
Moderate	88.3	87.9	96.6	98.4
Mild	88.2	88.0	96.4	99.0
Benign	87.6	87.3	96.3	98.2

Accuracy Comparison Analysis

Fig 8 represents a graphical comparison of accuracy across the four impact classes. The accuracy metric is used to evaluate the complete accuracy of a model's detections. In comparing the accuracy of different models the proposed ECBoA-OFS shows better results in the context of four impact classes. **Table 7** provides the average accuracy of the model's performance for each impact class.



(a) Severe

(b) Moderate

(c) Mild

(d) Benign

Fig 8. Accuracy Comparison Analysis for Different Attack Class.

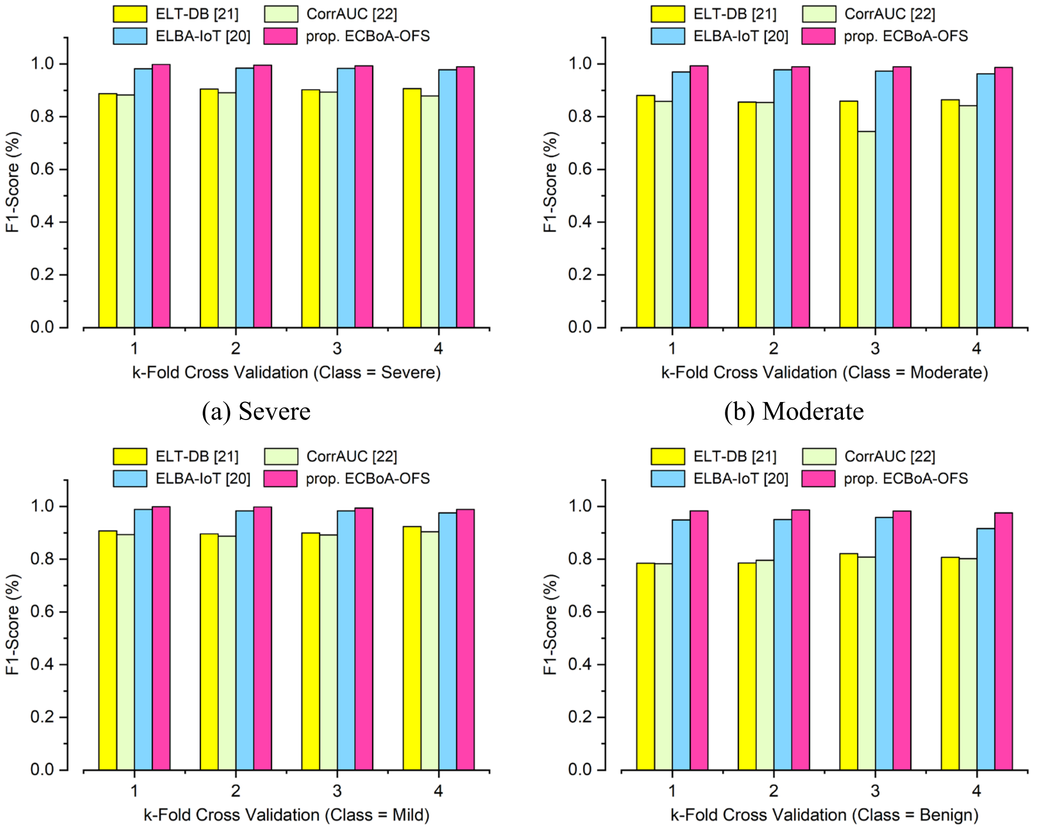
Table 7. Average Accuracy Result Comparison

Class	ELT-DB [21]	CorrAUC [22]	ELBA-IoT [20]	prop. ECBoA-OFS
Severe	87.1	86.5	97.5	99.7
Moderate	86.8	86.5	96.1	98.5
Mild	88.1	86.4	96.9	98.7
Benign	87.8	88.1	96.6	99.8

The accuracy of system attack detection measures the overall correctness of the identified samples of traffic. It signifies the ratio of accurately identified testers to the complete number of identified test data. It suggests that the proposed ECBoA-OFS has a high accuracy (close to 99.8%) in identifying benign traffic and a slightly lower accuracy (98.5%) in identifying moderate traffic. The accuracy values indicate how well the model performs in correctly identifying different types of traffic. The ECBoA-OFS method seems to be quite effective, especially in identifying benign traffic, and a slightly lower accuracy rate for other classes but still maintains a reasonable level of accuracy.

F-1 Score Comparison Analysis

Fig 9 presents a comparison analysis of F1-Scores for 4 impact classes. The F1-Score is a measure that utilizes precision and sensitivity for computation to consider a balanced assessment of a model's outcomes, especially in binary classification problems. The proposed ECBoA-OFS exhibits better results across all four impact classes compared to other approaches. This suggests that ECBoA-OFS achieves a better balance between precision and recall for the class prediction. **Table 8** provides the average performance for each impact class for a more granular understanding of how well the model performs across different categories in the environment of intrusion predictions.



(a) Severe

(b) Moderate

(c) Mild

(d) Benign

Fig 9. F1-Score Comparison Analysis for Different Attack Class.

Table 8. Average F1-Score Result Comparison

Class	ELT-DB [21]	CorrAUC [22]	ELBA-IoT [20]	prop. ECBoA-OFS
Severe	0.900	0.887	0.982	0.994
Moderate	0.865	0.825	0.971	0.990
Mild	0.907	0.894	0.983	0.995
Benign	0.800	0.797	0.943	0.982

The performance comparison of F1-Score between existing and proposed ECBoA-OFS models considers both precision and recall, making it useful for assessing a model's ability to balance true positive detections and minimize false positives. The comparison is conducted using a 4-fold cross-validation approach. The proposed ECBoA-OFS model achieved a maximal F1-Score of 0.995% with mild attacks and a lowest score of 0.982% with benign instances. A high F1 score (such as 0.995%) suggests that the ECBoA-OFS model is effective in detecting intrusions while keeping false alarms to a minimum. The results indicate that the ECBoA-OFS model performs well in intrusion detection, achieving high F1-Scores across different attack classes.

V. CONCLUSION

This paper presents an ensemble classification model named ECBoA-OFS, designed to enhance the prediction of botnet attacks. The key focus is on feature selection, utilizing fully optimized features from both traffic flow and botnet features.

A CPR technique is applied to build the most optimal feature selection. The ECBoA-OFS classification methodology aims to improve botnet attack detection by employing an ensemble approach that combines RF, AdaBoost, and KNN algorithms. The N-BaIoT-2021 dataset is utilized as a benchmark to evaluate the outcomes of the proposed ECBoA-OFS model in comparison to existing works. Data is split into training and testing for evaluation. The outcomes specify that the ECBoA-OFS attains impressive result scores, with an average precision of 99.6%, sensitivity of 99.3%, specificity of 99.0%, accuracy of 99.8%, and F1 score of 99.5%. Furthermore, the use of optimized feature selection and k-fold cross-validation is highlighted to validate the results of the ECBoA-OFS model. The comparison analysis suggests that the ECBoA-OFS outperforms state-of-the-art studies in terms of botnet attack prediction, emphasizing the significance of ensemble methods and feature optimization in enhancing the correctness and reliability of detection systems. In feature, it can advance models that can quickly adapt to new feature selections with minimal data by leveraging meta-learning techniques to adaptively balance training data based on real-time feedback during model training to different classes or instances.

Data Availability

No data was used to support this study.

Conflicts of Interests

The author(s) declare(s) that they have no conflicts of interest.

Funding

No funding agency is associated with this research.

Competing Interests

There are no competing interests

References

[1]. N. Islam et al., "Towards Machine Learning Based Intrusion Detection in IoT Networks," Computers, Materials &amp; Continua, vol. 69, no. 2, pp. 1801–1821, 2021, doi: 10.32604/cmc.2021.018466.

[2]. M. A. Rahman and A. T. Asyhari, "The Emergence of Internet of Things (IoT): Connecting Anything, Anywhere," Computers, vol. 8, no. 2, p. 40, May 2019, doi: 10.3390/computers8020040.

[3]. K. Zhao and L. Ge, "A Survey on the Internet of Things Security," 2013 Ninth International Conference on Computational Intelligence and Security, Dec. 2013, doi: 10.1109/cis.2013.145.

[4]. Y. K. Saheed and S. Misra, "A voting gray wolf optimizer-based ensemble learning models for intrusion detection in the Internet of Things," International Journal of Information Security, vol. 23, no. 3, pp. 1557–1581, Jan. 2024, doi: 10.1007/s10207-023-00803-x.

[5]. N. Pandey and P. K. Mishra, "Detection of DDoS attack in IoT traffic using ensemble machine learning techniques," Networks and Heterogeneous Media, vol. 18, no. 4, pp. 1393–1409, 2023, doi: 10.3934/nhm.2023061.

[6]. Z. Tian, C. Luo, J. Qiu, X. Du, and M. Guizani, "A Distributed Deep Learning System for Web Attack Detection on Edge Devices," IEEE Transactions on Industrial Informatics, vol. 16, no. 3, pp. 1963–1971, Mar. 2020, doi: 10.1109/tii.2019.2938778.

[7]. S. Nomm and H. Bahsi, "Unsupervised Anomaly Based Botnet Detection in IoT Networks," 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), Dec. 2018, doi: 10.1109/icmla.2018.00171.

[8]. H. Bahsi, S. Nomm, and F. B. La Torre, "Dimensionality Reduction for Machine Learning Based IoT Botnet Detection," 2018 15th International Conference on Control, Automation, Robotics and Vision (ICARCV), Nov. 2018, doi: 10.1109/icarcv.2018.8581205.

[9]. U. Inayat, M. F. Zia, S. Mahmood, H. M. Khalid, and M. Benbouzid, "Learning-Based Methods for Cyber Attacks Detection in IoT Systems: A Survey on Methods, Analysis, and Future Prospects," Electronics, vol. 11, no. 9, p. 1502, May 2022, doi: 10.3390/electronics11091502.

[10]. Y. Meidan et al., "N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders," IEEE Pervasive Computing, vol. 17, no. 3, pp. 12–22, Jul. 2018, doi: 10.1109/mprv.2018.03367731.

[11]. A. A. Alsulami, Q. Abu Al-Haija, A. Tayeb, and A. Alqahtani, "An Intrusion Detection and Classification System for IoT Traffic with Improved Data Engineering," Applied Sciences, vol. 12, no. 23, p. 12336, Dec. 2022, doi: 10.3390/app122312336.

[12]. M. Eskandari, Z. H. Janjua, M. Vecchio, and F. Antonelli, "Passban IDS: An Intelligent Anomaly-Based Intrusion Detection System for IoT Edge Devices," IEEE Internet of Things Journal, vol. 7, no. 8, pp. 6882–6897, Aug. 2020, doi: 10.1109/jiot.2020.2970501.

[13]. T. A. Alamiedy, M. Anbar, A. K. Al-Ani, B. N. Al-Tamimi, and N. Faleh, "Review on Feature Selection Algorithms for Anomaly-Based Intrusion Detection System," Recent Trends in Data Science and Soft Computing, pp. 605–619, Sep. 2018, doi: 10.1007/978-3-319-99007-1_57.

[14]. K. Albulayhi, Q. Abu Al-Haija, S. A. Alsuhibany, A. A. Jillepalli, M. Ashrafuzzaman, and F. T. Sheldon, "IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method," Applied Sciences, vol. 12, no. 10, p. 5015, May 2022, doi: 10.3390/app12105015.

[15]. F. Palmieri, U. Fiore, and A. Castiglione, "A distributed approach to network anomaly detection based on independent component analysis," Concurrency and Computation: Practice and Experience, vol. 26, no. 5, pp. 1113–1129, Jun. 2013, doi: 10.1002/cpe.3061.

[16]. U. M. Rao and J. Sastry, "Machine Intelligence by Central Pivot Ranges (MICPR): An Optimal Resource Scheduling Strategy for Cloud Services," Jun. 2022, doi: 10.21203/rs.3.rs-1632741/v1.

[17]. W. N. H. Ibrahim et al., "Multilayer Framework for Botnet Detection Using Machine Learning Algorithms," IEEE Access, vol. 9, pp. 48753–48768, 2021, doi: 10.1109/access.2021.3060778.

[18]. H. Alkahtani and T. H. H. Aldhyani, "Botnet Attack Detection by Using CNN-LSTM Model for Internet of Things Applications," Security and Communication Networks, vol. 2021, pp. 1–23, Sep. 2021, doi: 10.1155/2021/3806459.

[19]. J. L. Leevy, J. Hancock, T. M. Khoshgoftaar, and J. M. Peterson, "IoT information theft prediction using ensemble feature selection," Journal of Big Data, vol. 9, no. 1, Jan. 2022, doi: 10.1186/s40537-021-00558-z.

[20]. Q. Abu Al-Haija and M. Al-Dala'ien, "ELBA-IoT: An Ensemble Learning Model for Botnet Attack Detection in IoT Networks," Journal of Sensor and Actuator Networks, vol. 11, no. 1, p. 18, Mar. 2022, doi: 10.3390/jsan11010018.

[21]. A. Rezaei, "Using Ensemble Learning Technique for Detecting Botnet on IoT," SN Computer Science, vol. 2, no. 3, Mar. 2021, doi: 10.1007/s42979-021-00585-w.

[22]. M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, "CorrAUC: A Malicious Bot-IoT Traffic Detection Method in IoT Network Using Machine-Learning Techniques," IEEE Internet of Things Journal, vol. 8, no. 5, pp. 3242–3254, Mar. 2021, doi: 10.1109/jiot.2020.3002255.

[23]. S. D. A. Rihan , M. Anbar , and B. A. Alabsi, "Approach for Detecting Attacks on IoT Networks Based on Ensemble Feature Selection and Deep Learning Models," Sensors, vol. 23, no. 17, p. 7342, Aug. 2023, doi: 10.3390/s23177342.

[24]. B. Rosner, R. J. Glynn, and M. Ting Lee, "Incorporation of Clustering Effects for the Wilcoxon Rank Sum Test: A Large-Sample Approach," Biometrics, vol. 59, no. 4, pp. 1089–1098, Dec. 2003, doi: 10.1111/j.0006-341x.2003.00125.x.

[25]. E. C. Blessie and E. Karthikeyan, "Sigmis: A Feature Selection Algorithm Using Correlation Based Method," Journal of Algorithms &amp; Computational Technology, vol. 6, no. 3, pp. 385–394, Sep. 2012, doi: 10.1260/1748-3018.6.3.385.

[26]. M. G. Karthik and M. B. M. Krishnan, "Hybrid random forest and synthetic minority over sampling technique for detecting internet of things attacks," Journal of Ambient Intelligence and Humanized Computing, Mar. 2021, doi: 10.1007/s12652-021-03082-3.

[27]. T. T. Khoei, S. Ismail, and N. Kaabouch, "Boosting-based Models with Tree-structured Parzen Estimator Optimization to Detect Intrusion Attacks on Smart Grid," 2021 IEEE 12th Annual Ubiquitous Computing, Electronics &amp; Mobile Communication Conference (UEMCON), Dec. 2021, doi: 10.1109/uemcon53757.2021.9666607.

[28]. Y. Liao and V. R. Vemuri, "Use of K-Nearest Neighbor classifier for intrusion detection," Computers &amp; Security, vol. 21, no. 5, pp. 439–448, Oct. 2002, doi: 10.1016/s0167-4048(02)00514-x.

[29]. Q. A. Al-Haija and A. Ishtaiwi, "Multiclass Classification of Firewall Log Files Using Shallow Neural Network for Network Security Applications," Soft Computing for Security Applications, pp. 27–41, Oct. 2021, doi: 10.1007/978-981-16-5301-8_3.

[30]. T. Wu, Y. Hao, B. Yang, and L. Peng, "ECM-EFS: An ensemble feature selection based on enhanced co-association matrix," Pattern Recognition, vol. 139, p. 109449, Jul. 2023, doi: 10.1016/j.patcog.2023.109449.