IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 11 July 2022, accepted 23 August 2022, date of publication 5 September 2022, date of current version 14 September 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3204001

RESEARCH ARTICLE

In-Depth Feature Selection for the Statistical Machine Learning-Based Botnet Detection in IoT Networks

RAJESH KALAKOTI, (Graduate Student Member, IEEE), SVEN NÕMM, AND HAYRETDIN BAHSI
Department of Software Science, School of Information Technology, Tallinn University of Technology (TalTech), 12618 Tallinn, Estonia

Corresponding author: Rajesh Kalakoti (rajesh.kalakoti@taltech.ee)

This work was supported by the European Union through the European Social Fund through the Project Information Communication Technology (ICT) Program.

ABSTRACT Attackers compromise insecure IoT devices to expand their botnets in order to launch more influential attacks against their victims. In various studies, machine learning has been used to detect IoT botnet attacks. In this paper, we focus on the minimization of feature sets for machine learning tasks that are formulated as six different binary and multiclass classification problems based on the stages of the botnet life cycle. More specifically, we applied filter and wrapper methods with selected machine learning methods and derived optimal feature sets for each classification problem. The experimental results show that it is possible to achieve very high detection rates with a very limited number of features. Some wrapper methods guarantee an optimal feature set regardless of the problem formulation, but filter methods do not achieve that in all cases. The feature selection methods prefer channel-based features for detection at post-attack, communication, and control stages, while host-based features are more influential in identifying attacks originating from bots.

INDEX TERMS Feature selection, machine learning, Internet of Things, botnet, intrusion detection.

I. INTRODUCTION

IoT (Internet of Things) is shaping the way we live our human lives [1], from tiny toys to home-made applications to smart cities. IoT is a system of interrelated devices connected to the Internet to transmit and receive data from one device to other parts of the system; it can be an edge device, a cloud server, or another field device. At the same time, the IoT security issue has become more important as an enormous amount of data is associated with IoT networks. Due to the exponential growth of IoT devices [2], hackers and cybercriminals have more opportunities to exploit network vulnerabilities [3], resulting in various IoT-based botnet attacks [4], [5], [6]. The botnet, a large set of compromised machines controlled by attackers, is one of the strongest threats on the Internet to perpetrate cybercrimes, such as launching DDoS attacks [4], stealing sensitive data [7] or distributing malicious spam [8]. As a result, botnets act as a source of spreading malicious activity and usually threaten the availability of networks, in addition to other significant security consequences. It is important to develop security countermeasures against botnet threats.

A typical botnet life cycle has four phases, formation, command and control (C&C), attack and post-attack [9]. Attackers spread malware that helps them recruit new bots (that is, members of botnets) during the formation phase. C&C phase enables them to establish continuous communication with bots to control them for future actions. In the attack phase, attackers carry out malicious operations using bots. The post-attack phase covers activities related to the spread of IoT malware with the purpose of expanding the botnet. IoT networks constitute a lucrative target for botnet

The associate editor coordinating the review of this manuscript and approving it for publication was Chin-Feng Lai.

owners, as it is possible for them to recruit large numbers of IoT devices, which are usually shipped with various security vulnerabilities.

One of the effective security countermeasures against botnets is to establish security monitoring systems to detect malicious activities. An organization hosting various IoT devices is interested in the identification of devices that are compromised by IoT bot malware; therefore, its focus is much more on detection at formation, C&C or post-attack phases. On the other hand, organizations receiving attacks from IoT bots aim to prevent malicious traffic launched during the attack and post-attack phases. Therefore, it is important to develop a monitoring system that encompasses the entire botnet life cycle. This endeavor requires a more in-depth understanding of malicious actions and their characterization in each phase.

The Internet of Things (IoT) has received great attention in research on network anomalies and intrusion detection [10]. Malicious network traffic has been detected with conventional signature-based solutions such as Snort [11] or Suricata [12]. The drawback of signature-based systems is the inability to detect unknown or previously unidentified attacks, in addition to the obstacles that arise from mismanagement of signatures.

Instead of signature-based solutions, a behavior- or anomaly-based solution goes beyond identifying individual attack signatures to detect and analyze malicious behavior patterns. Machine learning is considered a viable solution that detects new variants of attacks with the elimination of the need for signatures. Although the application of statistical machine learning (ML) techniques has demonstrated highly accurate classification results in malicious traffic detection problems [13], feature selection as an important step in the ML workflow has not been fully addressed. The curse of dimensionality can be a concern that decreases detection performance due to overfitting when classifiers are trained with a large number of features [14]. In addition, a high-dimensional feature space may require more computing resources when the models are deployed in the operational environment. In most cases, intrusion detection systems should handle a large volume of network traffic, so maximizing resource usage is vital. IoT environments bring additional restrictions, so that detection sensors, system components that are responsible for the collection of network traffic and performing the detection function, may run on resource-constrained devices (e.g., edge devices). Therefore, reducing the size of the feature set can improve the performance of ML models in many ways. Additionally, feature selection helps to achieve a deeper understanding of the underlying approaches that rendered the data, since fewer features would be more perceivable by experts.

Various academic works [15], [16], [17], [18], [19], [20], [21] use feature selection techniques to improve the detection scores of existing ML classifiers. However, these studies do not explore the impact of feature selection methods on different binary and multiclass classification formulations that can be performed for intrusion detection at various stages of the botnet life cycle. More specifically, the set of features that is effective in detecting malicious traffic at one stage may not be instrumental at another stage. Furthermore, the performance of models that use different feature selection methods can vary according to the classification formulation.

The crux of this paper is to find the optimal subset of features with the help of filter and wrapper feature selection methods for various classification formulations that can be applied to IoT botnet attack detection. For this purpose, we have induced ML classifiers using the methods, extra tree classifier, random forest, decision tree, and k-nearest neighbor. The optimal feature sets are derived by a 10-fold cross-validation with classifiers from filter and wrapper methods.

In this research, we applied the feature selection methods to two datasets, namely N-BaIoT [22] and MedBIoT [23], which include network activities belonging to different steps of the botnet life cycle in IoT networks. Based on the phases of the botnet life cycle given in [9], we can deduce that N-BaIoT has instances related to the attack phase, while MedBIoT covers post-attack and C&C phases.

In addition to a binary classification, such as discriminating malicious traffic from benign traffic, it is possible to formulate various multiclass classification problems from these datasets. One of such formulations may focus on the detection of the malware type that induces the malicious traffic (e.g., Mirai, Bashlite), which is applicable for both datasets, whereas the second one may deal with the attack type that is conducted by the corresponding malware. For the latter case, N-BaIoT provides labels on the types of attacks that originated from infected devices (e.g., UDP flooding, spam), and MedBIoT has labels on whether the activity belongs to the C&C or post-attack phase. Depending on the situation, security administrators may be interested in different aspects of detection to make more informed operational decisions. For example, identifying the type of malware on the infected device would be necessary to apply the correct malware removal procedures. On the other hand, identifying the type of attack rather than the type of malware would be more essential for organizations that receive botnet attacks, as they need to develop defensive countermeasures to block or redirect network traffic accordingly. In our study, we investigate which feature sets are optimal for each binary and multiclass classification formulation and analyzed whether there exist variations in the optimal feature set that may impact the design considerations of intrusion detection in such different contexts. This contribution is unique because, to our knowledge, there is no study that provides a deeper analysis of the variations in feature sets that are effective in intrusion detection at different stages of the botnet life cycle.

The structure of this research work is described below. In Section II we have mentioned background work and a review of the literature related to botnet detection and feature selection. In Section III, the feature selection methods and experiments are described. Finally, our results are presented in Section IV. Section V gives a discussion of the main findings of this research work. Conclusions are drawn in Section VI.

II. BACKGROUND AND LITERATURE REVIEW

A. BOTNET DETECTION

Researchers have introduced traditional machine learning and data mining methods for botnet detection in recent decades and made significant advances. BotMiner [24], [25] and BotSniffer [26] used statistical algorithms to detect malicious traffic on an IoT network that is part of a botnet.

The Bayesian optimization Gaussian process (BO-GP) [27] is combined with the decision tree classifier as an optimized ML-based framework to detect botnet attacks on IoT devices. The detection rate for binary classification is improved to 99%.99 when the accuracy, precision, recall, and f1 score metrics are compared to the Decision Tree, SVM, with this optimized DT-BOGP framework. In this work, the Bot-IoT-2018 dataset [28] is used.

Convolutional neural networks (CNN) are used to detect IoT malware. This approach was created for the detection of Linux IoT botnets based on the PSI graph together with the CNN classifier [29]. Experiments were carried out using 4002 labeled IoT botnet datasets provided by the IoT-POT [30] team. These data sets were collected over one year, from October 2016 to October 2017. The detection rates, 92% precision and 94% F1 score, are achieved with the CNN classifier.

Yin et al. proposed the Bot-GAN framework to improve botnet detection performance [31]. Generative adversarial networks are used, where the GAN generator creates fake samples. A 3-layer LSTM network was selected as the generator and a 4-layer neural network architecture was chosen as the detector in the Bot-GAN setup. The ISCX dataset [32] is used for this framework. Of 491,381 training samples, 192,112 (39.10%) are malicious and include seven botnets, while the test set consists of 348,452 testing samples. This test set has 169988 (48.78%) malicious samples that possess 16 botnet types. The detector achieves 68.51% as an F1 score without having fake samples. The detector attains a maximum 70.59% of F1 score when the training set has 500 fake samples.

A hybrid deep learning scheme [33] is used to detect the botnet in the IoT network. A long-short-term memory autoencoder (LAE) is implemented to reduce the dimensionality of network traffic features. Then, the long-term interrelated network traffic behavior is analyzed with the help of bidirectional long-short-term memory (BLSTM) to achieve generalization ability. In this work, binary multiclassification problems are addressed in the BoT-IoT dataset [28] for the classification of network traffic. In general, 6 features were derived from 37 features of the dataset [28] with the help of the LAE and BLSTM classifier that achieved 100% precision, 93.17% MCC (Matthews correlation coefficient).

Alauthman *et al.* [34] have proposed a traffic reduction mechanism that integrates the reinforcement learning technique in three datasets. The first dataset is information security and objects technology (ISOT) that contains Storm Bot, Waledac Bot, and normal traffic. The second data set comprises four legitimate P2P applications (Vuze, uTorrent, Frostwire and eMule) and three P2P botnets (Zeus, Storm and Waledac) [35], and the third is the ISCX data set [32], which contains benign traffic. The authors have used real-world network traffic to evaluate their proposed approach and achieved a detection rate of 98.3% and a false positive rate of 0.012%.

Singh *et al.* [36] have developed a quasi-real-time intrusion detection system using open-source tools such as Hadoop, Hive, and Mahout to provide scalability for the identification of Peer-to-Peer botnet attacks. For this, the authors have built the packet capture module to process high data bandwidth in a quasi-real-time (within 5-30 s delay) and developed a distributed dynamic feature extraction framework to illustrate network traffic statistics of packet captures. The parallel processing power of Mahout (that is, a machine learning library built on top of Hadoop) was used to build the Random Forest model that achieved a detection performance of 99% precision and recall.

B. FEATURE SELECTION

Feature selection aims to find the best subsets of features from input data to achieve better prediction results by eliminating unnecessary features [37]. The feature selection methods were classified mainly into three categories, such as filter, wrapper, and embedded [14]. Filter methods utilize statistical methods to rank features according to their discriminatory power. They are usually applied in an initial step before inducing the models. However, wrapper methods use a machine learning model to evaluate the merits of a given set of features in terms of model performance to identify the optimal set. Embedded methods blend the advantageous factors of both the filter and wrapper methods so that they perform feature selection and training of the ML algorithm in parallel. This feature selection method is an integral part of the classification or regression model.

Many feature selection approaches have been applied to evaluate the importance of features related to the context of botnet detection. Entropy, impurity, RelieF and principal component analysis (PCA) [38] were used with the neural network classification algorithm. 99.20% detection rate was achieved with the top 10 features based on the entropy of a total of 29 features in two botnet datasets, ISOT [39] and ISCX [32].

Velasco-Mata *et al.* [40] has tested the feature sets 5, 6, 7 with two filter methods, Information Gain and Gini Importance, over Decision Tree, Random Forest, k-NN for botnet detection for multiclass classification. Finally, the set of five features produced an 85% detection rate with a decision tree classifier induced for the QB-CTU13 [41] and EQB-CTU13 [41] datasets.

Guerra-Manzanares *et al.* [19] proposed a hybrid approach by combining filter and wrapper methods with random forest and k-NN classifiers. Eighteen features are selected by

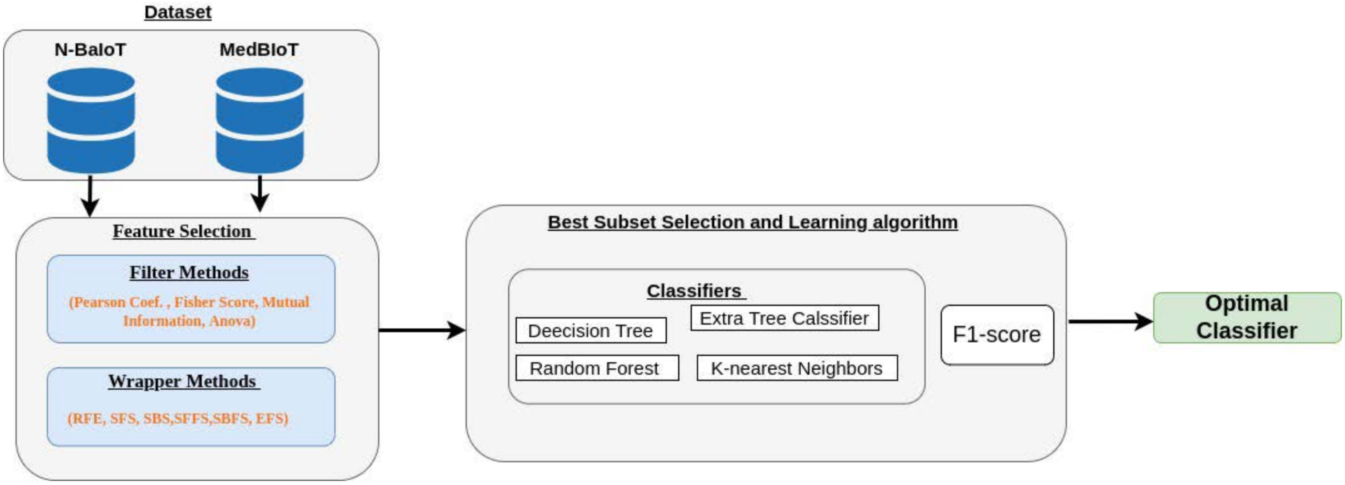


FIGURE 1. We used the filter and wrapper method feature selection approaches over the N-BaIoT [22] and MedBIoT [23] datasets to find the optimal feature subset.we evaluated all the feature subsets with four classifiers - DT, ET, RF and k-NN.

Pearson's correlation, and the top 20 features are selected with Fisher score. This study used the botnet dataset, N-BaIoT [22], which has 115 statistical features extracted from network traffic in an IoT network. The feature sets obtained from the filter methods are processed by wrapper methods, Sequential Forward Selection and Sequential Backward Elimination. Finally, a five-element set of features is used for the detection of IoT botnets formulated as a binary classification problem.

Correlation-based feature selection, consistency-based subset evaluation and principal component analysis [42] are used to select features that are then evaluated with decision trees, the Naive Bayes classifier, and the Bayesian Network classifier to detect botnet traffic based on peer-to-peer (P2P). With these selection methods, 5, 8, and 12 features were identified, respectively. 99% accuracy achieved with the decision tree based on the ISOT dataset [39].

Pektaş and Acarman [43] used linear models penalized with the L1 norm (also called Lasso), recursive feature elimination (RFE), tree-based feature selection methods (random forest feature importance) for the ISOT dataset [39]. Random forest feature selection produced 99% highest detection among all these feature selection methods.

The studies proposing feature selection do not create and compare the optimal sets that can be obtained for different multiclass problem formulations. In this paper, we address this gap by inducing various learning models for two datasets as explained in detail in Section II-C.

C. DATASET

In this study, we used two datasets, N-BaIoT [22] and MedBIoT [23]. Both datasets comprise legitimate IoT traffic as well as traffic with various types of attacks that originate from compromised IoT devices acting as bots.

N-BaIoT and MedBIoT have 115 and 100 features (mainly descriptive statistics measures), respectively, which are extracted from network traffic. These traffics are generated by bots deployed in a controlled testing environment. Both datasets have the same features, except that the MedBIoT dataset does not include network traffic coded as "H" in Table 1. More specifically, the features that are defined for each data point reflect the aggregated statistics of the raw streams of the network in five time windows (100 ms, 500 ms, 1.5 s, 10 s, and 1 min), which are coded L5, L3, L1, L0.1 and L0.01, respectively. There are five main feature categories, host-IP (traffic originated from a specific IP address, coded as H), host-MAC and IP (traffic originated from the same MAC and IP, coded MI), channel (traffic between specific hosts, coded HH), socket (traffic between specific hosts, including ports, coded HpHp), and network jitter (time interval between packets in channel communication, coded as HH_jit). For each major category, the packet count, mean and variance packet sizes are calculated. There have been extra statistical values like the correlation coefficient (PCC) of packet size, radius, covariance, magnitude, which are derived for Channel and Socket categories along with packet count, mean, variance. In this paper, we used a specific notation to name the features. The feature name is the concatenation of three keywords. The first one represents the category type (e.g., MI, HH), the second one shows the time window, and the third one indicates the statistical measurement function. For instance, "HH_L0.01_mean" means this feature is about the channel type that belongs to a 1-min interval with a mean function.

In this study, we have developed six different ML classification problems using these two datasets, as detailed in Table 2. The N-BIoT dataset is used for three classification problems, namely, binary, 3-class, and 9-class. Binary classification basically discriminates malicious traffic from benign traffic. 3-class provides greater scrutiny of malware type by classifying data points into categories, mirai, gafgyt, and benign. For the 9-class classification, the data points have been classified into different attack types: ack, benign,

TABLE 1. Summary of the features of the N-BaIoT and MedBIoT datasets features.

				N-BaIoT	MedBIoT
Feature Category	Category Code	Statistical Value Feature	Time Frame Window	No. of Features	No. of Features
Host Mac& IP	MI	Packet Count, Mean Variance	100 Micro Seconds	15	15
Host IP	H		500 Micro seconds	15	–
Network Jitter	HH_Jit		1.5 Seconds	15	15
Channel	HH	Packet Count, Mean Variance, Magnitude, Radius, Covariance, Correlation	10 Seconds	35	35
Socket	HpHp		1 Minute	35	35

TABLE 2. Classification problems addressed in this study.

Dataset	Classification Task	Class Name	Description of the Class name
N-BaIoT	Binary	Benign	Legitimate Network Traffic
		Attack	Malicious Network Traffic (Mirai, Gafgyt)
	3-class	Mirai	Mirai malware-infected network traffic
		Benign	Legitimate Network Traffic
		Gafgyt	Gafgyt malware-infected network traffic
	9-class	ACK	Gafgyt malware Sending Spam data
		Benign	Legitimate Network Traffic
		COMBO	Gafgyt malware Sending spam data and opening a connection to IP, port
		JUNK	Mirai Malware ACK-Flooding
		SCAN	Scans the network devices for vulnerabilities,(Mirai &Gafgyt)
		SYN	Mirai Malware SYN-Flooding
		TCP	Gafgyt malware TCP Flooding
		UDP	UDP flooding (Mirai & Gafgyt)
		UPDPLAIN	Mirai malwar UDP flooding with Less of an option for higher packet per second
MedBIoT	Binary	Benign	Legitimate Network Traffic
		Attack	Malicious Network Traffic (Mirai, Bashlite, Torii)
	3-class	Benign	Legitimate Network Traffic
		C&C	network traffic for C&C
		Spread	Spread Attack network traffic
	4-class	Bashlite	Bashlite malware-infected network traffic
		Benign	Legitimate Network Traffic
		Mirai	Mirai malware-infected network traffic
		Torii	Torii malware-infected network traffic

compact, junk, scan, syn, tcp, udp, and udpplain. These three-class and nine-class problem formulations address the attack phase of the botnet life cycle from two perspectives. The former identifies the types of malware that can be instrumental in detecting infected hosts in an organizational setting. The latter aims to discriminate against attacks carried out by bots, which better informs organizations that are targeted by such attacks.

MedBIoT is used for three classification formulations, binary, 3-class, and 4-class. As this dataset is collected at the C&C or formation phases, such formulations reveal which features are important in those phases. More specifically, 3-class addresses the identification of the phase (i.e., classes are benign, C&C and Spread), whereas 4-class aims to detect malware category (i.e., classes are benign, Bashlite, Mirai, and Torii).

In this work, we have experimented with 20,000 samples of each class label for the addressed classification type. For example, if the classification problem contains two classes, we randomly selected 40,000 samples from the source dataset.

III. FEATURE SELECTION METHODS

Within the framework of the present investigation, two types of feature selection methods are considered. The first is called the filter model, which evaluates a feature or a subset of features using a class-sensitive discriminating criterion [44]. These techniques do not depend on the particular classification algorithm. The second type of technique is the wrapper model. Techniques of this type use the characteristics of the specific classification algorithm to choose the feature set.

A. FILTER MODELS

In the domain of numeric feature sets, there are four main types of techniques. The first utilizes the linear correlations between the features. The second is based on the relationship between the inter-class and intra-class separation. The third uses entropy, and the fourth is based on the analysis of variance.

1) PEARSON's CORRELATION BASED TECHNIQUE

Based on Pearson's correlation coefficient (see (1)), the technique requires one to compute the collinearity matrix for the entire set of features to find the redundancy of the features. Pearson's correlation technique computes the linear correlation relationship between two variables. Pairwise correlations between features are analyzed to find the redundancy of features. $\mathcal{P}$-value of correlation coefficients bounds the ranges between -1 and 1. Two features contain a perfect positive correlation if the value is $\mathcal{P} = 1$. There is no correlation between the two features if the value $\mathcal{P} = 0$, and a perfect negative correlation is accepted if the value $\mathcal{P} = -1$. The formula for the Pearson correlation

$$\mathcal{P} = \frac{\sum_{i=1}^{n}[(x_i - \mu_x)(y_i - \mu_y)]}{\sqrt{\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\sum_{i=1}^{n}(y_i - \mu_y)^2}} \quad (1)$$

In (1), μ_x and μ_y denote the means of features x and y respectively. Greater absolute values of Pearson's correlation coefficient indicate stronger linear dependence between the features.

2) FISHER SCORE

Fisher score [44] is designed for the numeric features and measures the ratio of the average inter-class separation to the average intra-class separation. It is also referred to as Fisher's ratio [45]. Formally defined in (2) and denoted as F_s (not to be confused with F1 score), the numerator calculates the average inter-class separation and, the denominator calculates the average intra-class separation.

$$F_s = \frac{\sum_{j=1}^{K} p_j(\mu_j^i - \mu^i)^2}{\sum_{j=1}^{K} p_j(\sigma_j^i)^2} \tag{2}$$

where μ_j^i and σ_j^i are the mean and standard deviation of the j-th class and i-th feature, p_j is the proportion of data points of class belonging to the class j. Greater Fisher's score values indicate greater discriminating power of the feature.

3) MUTUAL INFORMATION

Among the different techniques implementing mutual information exclusion idea *normalized mutual information feature selection* [46] was chosen. For the case of continuous variables mutual information (MI) is defined by [46] as follows:

$$I(\mathcal{X}, Y) = \int\int p(x, y) \log \frac{p(x, y)}{p(x), p(y)} dxdy \tag{3}$$

Here, $p(x, y)$ is the joint probability density function (PDF) of the variables $\mathcal{X}$, Y and $p(x)$ and $p(y)$ are the marginal PDFs of the respected variables. For the case of discrete variables, [46] defines MI as follows:

$$I(X; Y) = \sum\nolimits_{y \in Y} \sum\nolimits_{x \in X} p(x, y) \log \frac{p(x, y)}{p(x)p(y)}. \tag{4}$$

In (5) $p(x, y)$ denotes the joint probability mass, the function, the function, and $p(x)$ and $p(y)$ are the marginal probabilities. Mutual information values fall in the interval given below.

$$0 \leq I(X; Y) \leq \min\{H(X), H(Y)\} \tag{5}$$

To make this paper self-sufficient, the main steps of the MI -based feature selection algorithm proposed by [46] are presented below. Denote $I(C; S)$ the MI between the class variable C and the subset of selected features S. Also define measure G as

$$G = I(C; f_i) - \frac{1}{|S|} \sum_{f_s \in S} NI(f_i; f_s). \tag{6}$$

1) Initialize the initial feature set F that includes all available features and the empty set S of the selected features.
2) Calculate $I(f_i, C)$ or each feature $f_i \in F$.
3) To select the first feature, find $\hat{f_i}$ such that $\hat{f_i} = \max_{i=1,\ldots,N}\{I(f_i, C)\}$.
4) Update sets F and S as follows: $F = F \setminus \hat{f_i}$ and $S = \hat{f_i}$
5) Repeat until $|S| = k$.
    a) Compute $I(f_i; f_s)$ for all pairs of features such that $f_i \in F$ and $f_s \in S$.
    b) Select the feature $f_i \in F$ that maximizes the measure (6).
    c) Update sets F and S as follows: $F = F \setminus \hat{f_i}$ and $S = \hat{f_i}$
6) Return the set S

4) ANOVA F-TEST

ANOVA is one of the most well-known feature selection techniques, therefore, does not require an in-depth explanation. This method usually answers the question of whether the values of the given features are independent of the target classification label or not. It is performed in the form of statistical hypothesis testing, where the null hypothesis states that the values of the feature are independent of those of the target label, and the alternative hypothesis states the opposite. The application of this method requires the user to utilize only the features whose values are not independent of the target labels.

B. WRAPPER METHODS

Unlike the filter models, wrapper methods are classifier-agnostic and choose the most suitable feature set for the particular classifier. The wrapper method is used to calculate the weights of the features using the classification algorithm to measure the performance of the features. Wrapper methods employ the inductive algorithm as an evaluation or criterion function [47], [48]. This approach uses a classification algorithm to evaluate subsets of features based on their predictive accuracy (in test data) after cross-validation of the dataset. In the context of our research, we have evaluated subsets of features using the F1 score. Usually, the feature set is being constructed iteratively by adding (forward selection) or deleting (backward elimination) the features. Within each branch, particular methods differ by evaluating the significance of the features, the goodness criteria of the model, and the number of features added or removed. In the preliminary stage of the investigation, the authors have experimented with six different wrapper techniques. Among them, Recursive Feature Elimination (RFE) [49], Sequential Backward Selection (SBS), and Sequential Forward Selection (SFS) [50] have shown the best results and are included in the comparison.

1) RECURSIVE FEATURE ELIMINATION

Recursive feature elimination (RFE) is a greedy algorithm based on feature ranking techniques [49]. Based on a characteristic of the feature-ranking criterion, the RFE starts with a complete set of features and then removes the least relevant feature one by one to choose the most significant features. The RFE is used with the following classification algorithms, DT, ET, and RF. This method uses the following steps to evaluate the significance of the features.

1) Initialize the initial set of features F that includes all available features, set each element of the feature ranking list R to $1/n$.
2) Repeat the following steps until the feature set $F = \emptyset$
   - Train with the classification algorithm and calculate the importance of the feature in set F. Order the features corresponding to their importance and update the list R accordingly.
   - Eliminate the feature of the smallest importance.
3) Output: List of Feature Rankings R.

2) SEQUENTIAL FORWARD SELECTION

We have used two sequential algorithms [50] that work based on greedy search algorithms. SFS [50] is a stepwise search approach that can avoid excessive computational time consumption. It works in a bottom-to-top approach. The following steps are involved in the SFS Algorithm.

1) Start with an empty set $S = \emptyset$, $F = f_1, f_2, \ldots .f_n$
2) **while** |F|>0
   # |F| is size of the feature set F
3) $f_i = argmin_{j \in F}[J(S + f_i)]$
   (Select the feature $f_i \in F$ with the maximum performance of the classification algorithm and join to the set S (the features selected subsequently combine with the initial selected feature)
4) $S = S + f_i$
5) $F = F - f_i$

Consider F to be a set of features. Then select the best feature among the F features using some evaluation criterion function J that maximizes the performance of the classification algorithm. The F1 score is considered an objective evaluation criterion function. At each iteration, a new feature subset is created with the help of one of the remaining available features and the previous feature subset. The new subset of features should provide the maximum classification performance compared to the addition of any other feature. This iteration continues until the total number of features is completed in the set F. SFS method is the best and most rapid method when a small subset of optimal features is available.

3) SEQUENTIAL BACKWARD SELECTION

In contrast to SFS, SBS (Sequential Backward Selection) operates in a top-to-bottom approach. The selection of features starts from a set F with n being the total number of features. Therefore, the evaluation function produces the maximum performance of the classification algorithm for all n numbers of features. Each feature is removed one at a time. For every iteration, the new subset is created by the $n - 1$ features computed with the help of the evaluation function, and then the worst feature is discarded from the next subset of features. This procedure continues until the total number of features is left.

1) S = feature set, $F = f_1, f_2, \ldots .f_n$
2) **while** |F|>1 do
   #|F| is size of the feature set F,
3) $f_i = argmin_{j \in F}[J(S - f_i)]$
4) $S = S - f_i$
5) $F = F - f_i$

TABLE 3. Tuning of learning algorithm hyperparameters.

Algorithm	Description	Range
	Maximumn Depth of the tree	5-50 in steps of 1
Decision Tree	Minimum number of samples required to split an internal node	2-30 in steps of 1
	the minimum number of samples required to be at a leaf node.	2-30 in steps of 1
	the impurity of a split	Gini, Entropy
	Maximum number of levels in each decision tree	5, 500 in steps of 50
	Maximum number of features	Square root, Auto ,
Random Forest	Maximumn Depth of the tree	5-50 in steps of 1
Extra Tree Classifier	the minimum number of samples required to be at a leaf node.	2-30 in steps of 1
	the minimum number of samples required to be at a leaf node.	2-30 in steps of 1
	the impurity of a split	Gini, Entropy
k-nearest neighbors	number of neighbors	1-25 in steps of 1
	Distance metric	minkowski,euclidean,manhattan

C. APPLICATION OF THE MACHINE LEARNING WORKFLOW

For the computational experiments, the classical machine learning workflow was used. The initial datasets are large enough to provide samples that can be balanced with respect to all characteristics of the dataset, malware type, attack type, and device type. In the preprocessing step, balanced samples were drawn from the dataset of interest. Then, the division into training and testing subsets was carried out proportionally 80/20. Initial experiments have demonstrated that among the k-nearest neighbors classifier (kNN), decision tree classifier (DT), random forest classifier (RF), extremely randomized trees classifier (ET), logistic regression, support vector machine, and Ada-boost classifier, the last three have demonstrated much lower performance and were excluded from further investigation. For each remaining classifier and feature selection technique, a ten-fold cross-validation was performed, while, to ensure better results and the best configuration for each classification algorithm, a randomized search was used to find the optimal hyperparameters for each classifier. The range of hyperparameters is described in Table 3.

We use the three steps to evaluate the distinct subsets of features in both datasets. First, the F1 score metric is used to evaluate the set of features. Second, computational time is the total time it takes a computer with a particular processor to complete a task. Third, Performance computed the ratio between the F1 score and the computational time. Intrusion detection systems must respond as quickly as possible without sacrificing accuracy. Response time is essential when thwarting the threat in the early stages would limit the degree of losses. For this motivation, time must be considered when evaluating any detection of the model along with the model metrics. The F1 score (see Eq. (7)) is defined as a harmonic mean of precision (P) and recall(R) [51]. In this research work, precision is the fraction of correctly identified botnet samples to all botnet samples identified as a botnet.

TABLE 4. Filter method feature sets for the N-BaIoT and MedBIoT dataset.

Dataset	Classificaiton type	Pearson Correlation	Fisher Score	Mutual Information	Anova
N-BaIoT	Binary	33	5	3	3
	3-class	33	6	3	5
	9-class	33	68	28	59
MedBIoT	Binary	34	51	36	85
	3-class	34	42	38	49
	4-class	34	46	41	52

TABLE 5. Wrapper methods feature sets for N-BaIoT and MedBIoT.

Dataset	Classification Type	Feature Selection Approach	DT	RF	ET	Knn
N-BaIoT	Binary	RFE	3	4	4	
		SFS	3	3	3	3
		SBS	3	3	3	3
	3-class	RFE	3	4	4	
		SFS	3	3	3	3
		SBS	3	3	3	3
	9-class	RFE	28	23	25	
		SFS	3	3	3	3
		SBS	3	3	3	3
MedBIoT	Binary	RFE	29	27	24	
		SFS	7	7	7	7
		SBS	7	7	7	7
	3-class	RFE	26	27	24	
		SFS	7	7	7	7
		SBS	7	7	7	7
	4-class	RFE	29	24	22	
		SFS	7	7	7	7
		SBS	7	7	7	7

On the other hand, recall is the fraction of correctly identified botnet samples for all botnet samples in the dataset [52]. The F1 score provides a more suitable measure of incorrectly classified cases than the accuracy measure. We have used the harmonic mean of the F1 score, as it penalizes the extreme values. F1 score as follows;

$$\text{F1 score} = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{7}$$

In our experiments, we used the computational time to calculate the computational cost of classifying a sample. We did not consider the training time of the ML algorithms. We have experimented with all tasks on the same CPU. Finally, to measure the performance of a set of features derived from filter and wrapper methods, we calculated the ratio between the F1 score and the computational time to allow measurement of the gain in detection ability relative to the computational expense of this detection [40].

The experiment carried out in this work was carried out on a Ubuntu 20.04.4 LTS machine with 60 GB of DDR4-2666 R ECC RAM and 2 x Intel Xeon Gold 6148 20C 2.40 GHz. We developed our scripts using Python 3, Scikit-learn [53] and mlextend libraries [54].

IV. RESULTS

This section gives experimental results of the learning models induced for six classification problems listed in Table 2. We analyze the importance of the features obtained by filter and wrapper feature selection methods in each problem and perform a comparison between the results. Tables 4 and 5 show the numbers of features selected by the filter and wrapper methods for each classification problem, respectively. We provided detailed analysis of the result of each classification problem in the following subsections.

A. N-BaIoT

1) BINARY CLASSIFICATION

In this part, we use filter and wrapper feature selection methods to find the optimal feature subsets for binary classification of the N-BaIoT dataset. Based on the ratio of the highest detection rate of the minimal feature set to its computational time, as given in Fig. 2, we selected the best model for the implementation of four classifiers with different feature selection methods. In this binary problem formulation, we identified 33 features with fewer correlations according to Pearson's correlation values. For each filter method, we select the best features based on their scores. Furthermore, we induce models with feature sets that have increasing numbers to understand how many features are enough to pass the 99% F1 score. Finally, we select the best 3, 5, 3 features for the ANOVA, Fisher Score and mutual information methods, respectively (see Table 4). On the other hand, the wrapper methods usually select three features (for example, DT selects three features in each method), as presented in Table 5.

Almost all classifier and feature pairs produce a high detection rate above 99%, as shown in Table 6. Based on the minimal set and computational performance, we selected three pairs and reported more detailed performance results, accuracy, precision, recall, and F1 score values in Table 7. These pairs are: DT with mutual information (that is, three features), Fisher (that is, five features), and SBS (that is, three features). DT with SBS achieves the highest performance metric, as shown in Fig. 3. Among the wrapper methods, Anova provides better results than the others.

The mutual information method selected the features, {MI_dir_L0.1_weight, MI_dir_L0.01_weight, H_L0.01_ weight}, fisher score selected {MI_dir_L5_weight, HH_jit_L5_mean, MI_dir_L5_mean, MI_dir_L0.01_weight, MI_dir_L0.01_mean}, Anova identified the feature set, {MI_dir_L1_weight, MI_dir_L0.1_weight, H_L0.1_weight}. SBS selected {MI_dir_L5_weight, MI_dir_L3_weight, MI_dir_L1_ weight}. Almost all features belong to the host category; except one that is a network jitter-type feature.

It is important to note that we computed the computational time of the models (i.e. the testing-time performance) after selecting the features in all filter and wrapper methods. Thus, the time required for feature selection is not reported in this paper, as testing time is a more significant aspect compared to training, which is not done so frequently, and, when needed, high resources can be assigned for such task. In this sense, the calculated time can be affected by the number of features and characteristics of the corresponding learning model. However, in our experiments, as expected, we observed that

TABLE 6. F1 scores for binary classification models using feature subsets (represented in Table 4 and 5) of feature selection algorithms in the N-BaIoT dataset.

Binary N-BaIoT					
FS Method	**Approach**	**DT**	**ET**	**RF**	**KNN**
Filter	Pearson Correlation	0.9987	0.9983	0.9983	0.9957
	Fisher Score	0.9997	0.9997	1.0000	0.9847
	Mutual Information	0.9990	0.9990	0.9990	0.9977
	Anova	0.9973	0.9970	0.9970	0.9970
Wrapper	RFE	0.9983	0.9983	0.9987	
	SFS	0.9996	1.0000	0.9999	0.9994
	SBS	0.9998	0.9999	0.9997	0.9966

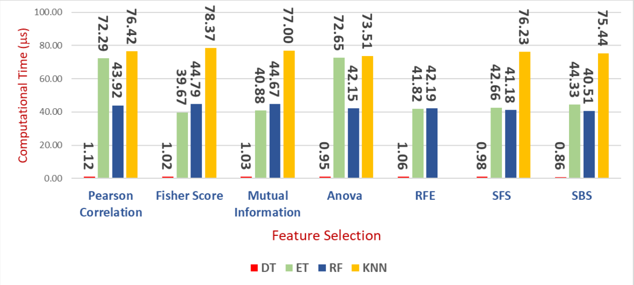


FIGURE 2. Computational time required to classify a sample by binary classification models on N-BaIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

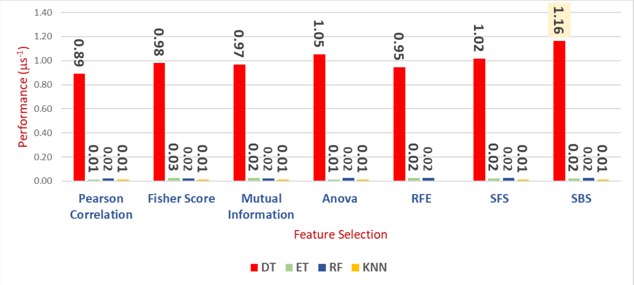
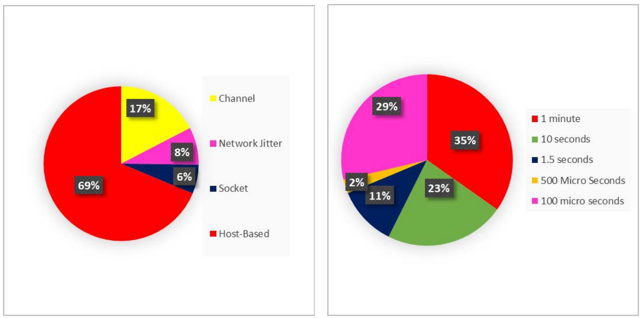


FIGURE 3. Performance achieved by binary classification models over the N-BaIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

TABLE 7. Accuracy, precision, recall, F1, Binary classification scores of the selected model with performance based on feature sets in the N-BaIoT dataset.

Feature Selection	Feature Subset	Model	Class Name	Accuracy	Precision	Recall	F1-Score
Mutual Information	3	DT	Attack	0.99	0.99	0.99	0.99
			Benign	0.99	0.99	0.99	0.99
Anova	3	DT	Attack	0.99	0.99	0.99	0.99
			Benign	0.99	0.99	0.99	0.99
SBS	3	DT	Attack	1	1	1	1
			Benign	1	1	1	1



(a) Feature Category Contribution to Overall Botnet detection

(b) Time Window Contribution to overall botnet detection

FIGURE 4. Contribution of feature categories and time windows in selected feature sets for binary classification in the N-BaIoT dataset.

TABLE 8. F1 scores for 3-class classification models using feature subsets (shown in Table 4 and 5) of feature selection algorithms on the N-BaIoT dataset.

3-Class N-BaIoT					
FS method	**Approach**	**DT**	**ET**	**RF**	**KNN**
Filter	Pearson Correlation	0.9989	0.9989	0.9987	0.9338
	Fisher Score	0.9989	0.9989	0.9987	0.9338
	Mutual Information	0.9991	0.9989	0.9989	0.9730
	Anova	0.9965	0.9956	0.9921	0.9910
Wrapper	RFE	0.9991	0.9964	0.9904	
	SFS	0.9997	0.9997	0.9998	0.9970
	SBS	0.9996	0.9998	0.9994	0.9790

training the wrapper models is more computationally expensive compared to filter methods. Among the wrapper methods, sequential feature selection algorithms (SBS, SFS) are more expensive than recursive feature elimination.

After identifying the optimal feature subsets from the dataset for binary classification, we performed a frequency analysis to scrutinize which feature category and time windows are used primarily by the selection methods, as shown in Fig. 4. Host-based feature categories are observed to play an important role in discriminating malicious traffic from benign traffic. The features of network jitter and socket are less preferred. Although the features regarding the longest time window, 1 minute, have contributed greatly to the detection, there is no clear increasing or decreasing pattern regarding the time duration, as the shortest duration, 100 microseconds, also plays a significant role in the model performance.

2) 3-CLASS CLASSIFICATION

In the N-BaIoT dataset, Mirai and Gafgyt malware are used to infect IoT devices. In this part, we report the findings of the three-class classification models that discriminate network traffic as Mirai, Gafgyt, and legitimate. Similarly, we evaluated the feature selection method and the pairs of learning models according to the same performance metric we used for binary classification and presented the F1 scores in Table 8.

All pairs, except some KNN models, provide more than 99% F1 scores. Pearson correlation still found 33 features. We identified six, three, and five features by using filter methods, fisher score, mutual information, and ANOVA,

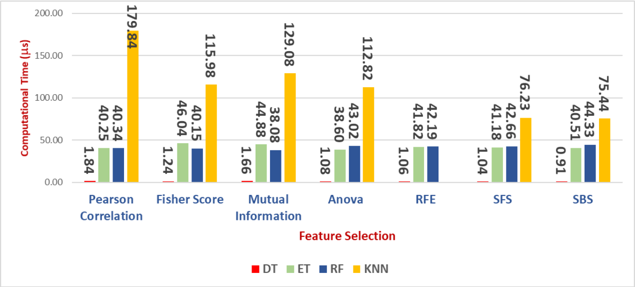


FIGURE 5. Computational time required to classify a sample using 3 class classification models in the N-BaIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

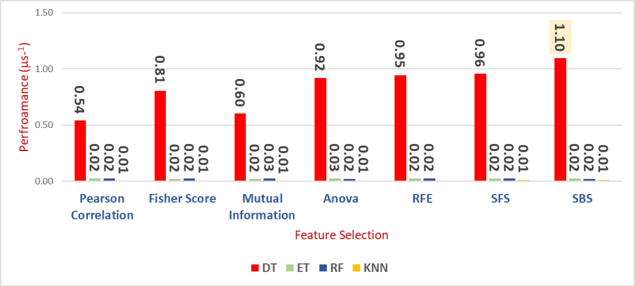


FIGURE 6. Performance achieved by 3-class classification models in the N-BaIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

respectively, as shown in Table 4. The wrapping methods mostly selected three features (see Table 5).

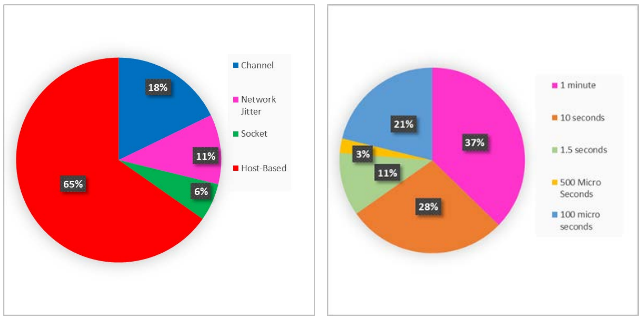
Among all the feature selection methods, the DT and SBS pair again achieves the highest performance, as shown in Fig. 6. Anova is the best compared to other filter methods. Table 9 shows the detailed performance metrics for DT and three feature selection methods, Fisher Score, Mutual Information, and SBS. It is obvious that the detection performance is higher than 99% for all metrics.

The optimal feature set selected by the mutual information feature set is {MI_dir_L0.1_mean, MI_dir_L0.01_mean, H_L0.01_mean}, the set of Fisher Score is { MI_dir_L5_weight, MI_dir_L5_mean, MI_dir_L0.01_mean, MI_dir_L0.01_weight, H_L0.01_mean}.

Compared to binary classification, we were unable to identify clear differences between the results. Learning models can easily identify the type of malware in this dataset. However, a small number of features, 3-5, achieve high detection rates regardless of the feature selection method. SBS and DT are the pair that performs best. The analysis of category distributions for the classification of 3 classes is given in Fig. 7. The results are very similar to those of binary classification. Host-based features have again played an essential role, and the time-window distribution does not show a distinct outcome.

TABLE 9. Accuracy, Precision, Recall, F1 of 3-class classification of the selected model with feature set-based performance over the N-BaIoT dataset.

Feature Selection	Feature Subset	Model	Class Name	Accuracy	Precision	Recall	F1-Score
Fisher Score	5	DT	mirai	0.99	0.99	0.99	0.99
			benign	0.99	0.99	0.99	0.99
			gafgyt	1	1	1	1
Mutual Information	3	DT	mirai	0.99	0.99	0.99	0.99
			benign	0.99	0.99	0.99	0.99
			gafgyt	1	1	1	1
SBS	3	DT	mirai	0.99	0.99	0.99	0.99
			benign	0.99	0.99	0.99	0.99
			gafgyt	0.99	0.99	1	0.99



(a) Feature Category Contribution to Overall Botnet detection

(b) Time Window Contribution to overall botnet detection

FIGURE 7. Contribution of feature category and time window in the selected feature set for 3-class classification in the N-BaIoT dataset.

TABLE 10. F1 scores for 9-class classification models using feature subsets (see in Table 4&5) of feature selection algorithms in the N-BaIoT dataset.

9-class N-BaIoT					
FS Method	Approach	DT	ET	RF	KNN
Filter	Pearson Correlation	0.9946	0.9949	0.9944	0.7051
	Fisher Score	0.9955	0.9954	0.9941	0.9611
	Mutual Information	0.9937	0.9948	0.9912	0.9453
	Anova	0.9956	0.9952	0.9925	0.9601
Wrapper	RFE	0.9943	0.9962	0.9967	
	SFS	0.9927	0.9941	0.9942	0.9928
	SBS	0.9944	0.9947	0.9940	0.8502

3) 9-CLASS CLASSIFICATION

In the 9-class formulation, we consider eight different types of attack and benign as distinct categories, as presented in Table 2. The results of this classification are quite different from the results of the binary and 3-class classification with respect to filter methods, as the learning models with these selection methods require a very high number of features to achieve an F1 score greater than 99%. More specifically, 68, 28 and 59 features should be fed into the models when Fisher score, mutual information, and ANOVA methods are used, respectively. However, 33 features are identified as not highly correlated by the Pearson correlation method. Wrapper methods show very interesting results. Although RFE provides higher detection results using 20-28 features depending on the type of learning model, SFS and SBS achieved higher detection with only three features.

Table 10 shows the F1 scores achieved by the nine sets of classification features of the classes. Except for KNN, all

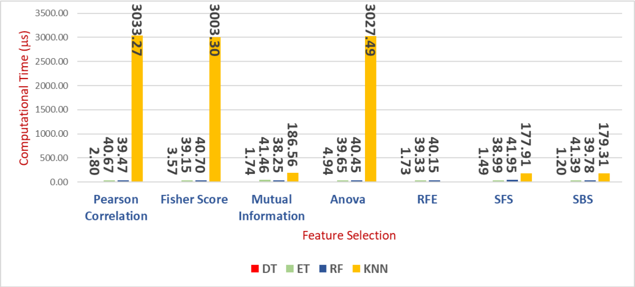


FIGURE 8. Computational Time required to classify a sample using 9-class classification models on the N-BaIoT dataset using feature sets (see Table 4 and 5) of feature selection methods.

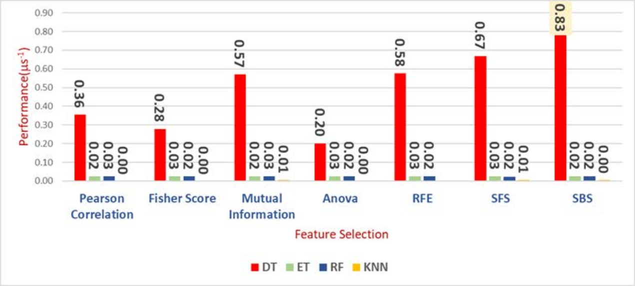


FIGURE 9. Performance achieved by 9-class classification models in the N-BaIoT dataset using feature sets (in Table 4 and 5) of feature selection methods.

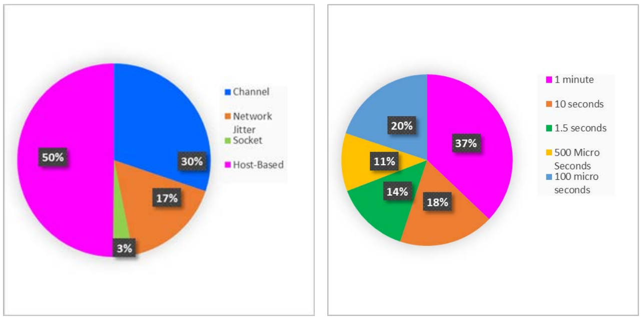
TABLE 11. Accuracy, Precision, Recall, F1 summary of classification of results mutual information and SBS features, DT with 28-feature set and 3-feature set respectively for 9-class classification over N-BaIoT dataset.

	Accuracy		Precision		Recall		F1-Score	
Class Name	MI	SBS	MI	SBS	MI	SBS	MI	SBS
Ack	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Benign	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99
Combo	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Junk	0.98	0.98	0.99	0.99	0.99	0.99	0.99	0.99
Scan	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
Syn	0.99	1	1	0.99	0.99	0.99	0.99	0.99
TCP	1	1	0.98	0.98	0.99	0.99	0.99	0.99
UDP	0.98	0.98	0.99	0.99	0.98	0.98	0.98	0.98
UDPPLain	0.99	0.99	1	1	0.99	0.99	0.99	0.99

other models achieve more than 99% in all selection methods. The result of the overall performance metric indicates that SBS and DT are the best pair in the 9-class classification (see Fig. 9). Among the wrapper methods, DT and mutual information emerge as the leading performer.

3 features used by the SBS and DT pair are as follows: MI_dir_L0.01_mean, HH_L0.01_std, HH_jit_L0.01_mean.

Table 11 shows the detailed classification performance of the 9-class classification with mutual information based on the 28-feature set and the SBS with the 3-feature set (that is, DT is the learning model in both cases). Although the detection rates of some classes (e.g., junk accuracy, accuracy, recall and F1 UDP scores) decrease to 98%, the remaining metrics show figures equal to or greater than 99%.



(a) Feature Category (b) Time Window

FIGURE 10. Feature category and time window in each set of features for 9-class classification.

The frequency analysis of the feature categories shows that the host-based features are still the most important category for the 9-class classification (see Fig. 10). However, the selected features of the channel category are higher compared to the binary and 3-class formulations. The contribution of the network jitter category is also more important in this classification task. This means that learning models need to resort to other features, which provide statistics about network activities between hosts and time intervals between network packets to differentiate attack types. When many types of attack are considered, including various denial-of-service attacks, such features are instrumental in making a distinction between them. Time window analysis provides a similar distribution, except that lower time intervals (i.e. 1.5 seconds, 500 microseconds, and 100 microseconds) have closer distributions to each other.

4) THE STANDARD FEATURE SET FOR BINARY, 3-CLASS AND 9-CLASS CLASSIFICATIONS OVER N-BaIoT

In this part, our objective was to discover a feature set that provides high performance for all classification models induced with the N-BaIoT dataset. Here, we do not claim to obtain the feature set that has been proven to be the best for all formulations, but we show that a working set is possible. Intuitively, for this purpose, we have tested the best feature sets of each classification in the other classification tasks. The best feature set obtained from the 9-class classification provided high detection rates for the remaining binary and 3-classification tasks. However, we were unable to obtain such high results in the reverse situation where binary or 3-class classification features are applied to a 9-class formulation. More specifically, the feature set, {MI_dir_L0.01_mean, HH_L0.01_std, HH_jit_L0.01_mean} that is determined by the SBS and DT pair for the 9-class classification is utilized to induce models for all classification types, and we obtained the results given in Table 12. Except for the Junk and UDP classes in the 9-class formulation, all results are equal to or greater than 99%, demonstrating the effectiveness of this common set in all classification types.

TABLE 12. Classification results using the standard 3-Feature Set for all classification tasks in the N-BaIoT dataset.

Classification Type	Class Name	Accuracy				Precision				Recall				F1-Score			
		DT	RF	KNN	ET	DT	RF	KNN	ET	DT	RF	KNN	ET	DT	RF	KNN	ET
Binary	Benign	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Attack	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
3-class	Mirai	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Benign	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Gafgyt	0.99	0.99	0.99	0.99	0.99	1	1	1	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99
9-class	Ack	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Benign	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Combo	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
	Junk	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	0.98	1
	Scan	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	1	0.99	1	0.99	0.99	0.99	1
	Syn	0.99	0.99	0.99	0.99	0.99	1	1	1	1	0.99	0.99	0.99	0.99	1	0.99	0.99
	TCP	0.99	0.99	0.99	0.99	0.99	1	1	1	0.99	0.99	1	0.99	0.99	0.99	1	0.99
	UDP	0.98	0.99	0.99	0.99	0.99	0.97	0.99	0.99	0.97	0.99	0.99	0.99	0.99	0.98	0.99	0.98
	UDPPLain	0.99	0.99	0.99	0.99	0.99	0.99	1	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99

TABLE 13. Classification performance of sequential back- ward selection, DT With 7-feature set for binary classification over MedBIoT dataset.

Feature Selection Method	Feature Set	Model	Class Name	Accuracy	Precision	Recall	F1-Score
SBS	7	DT	Attack	0.99	0.99	0.99	0.99
			Benign	0.99	0.99	0.99	0.99

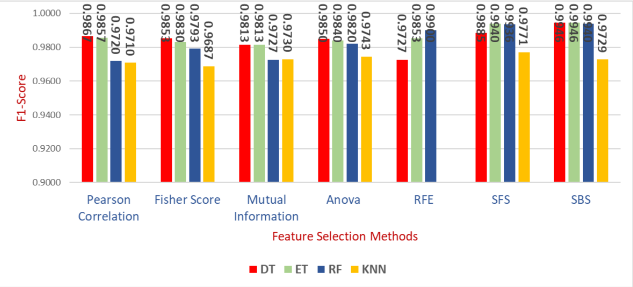


FIGURE 11. F1-scores for binary classification models in the MedBIoT dataset using feature subsets (see in Table 4&5) of feature selection algorithms.

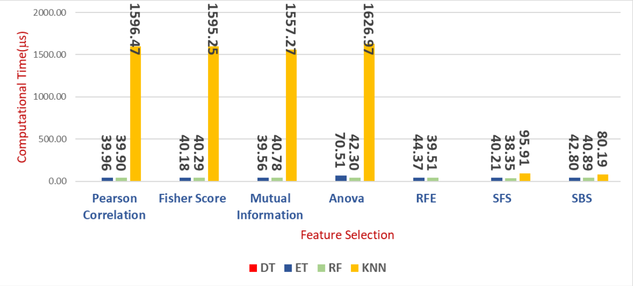


FIGURE 12. Computational time required to classify a sample by binary classification models over MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

B. MedBIoT

MedBIoT dataset has malicious network traffic from Mirai, BashLite, and Torii botnet malware, which were deployed on 83 real or emulated IoT devices. In this subsection, we report the experimental results of the binary, 3- and 4-class classification models induced with this data set (see Table 2 for the details of classification formulations).

1) BINARY CLASSIFICATION

We identified that 34 features are not highly correlated according to Pearson's correlation scores in the MedBIoT data set. A high number of features are required for filter methods to achieve a reasonable detection threshold rate equal to or above 98%. More specifically, ANOVA, Fisher Score, and Mutual Information can achieve that threshold rate with 85, 51, and 36, respectively, as shown in Table 4. On the other hand, RFE reaches the threshold value of 24-27 features depending on the type of learning model, while 7 features are enough for SBS and SFS (see Table 5). We present the F1 scores for all model and feature selection pairs in Fig. 11. Although the pairs do not exceed 98%, at least one learning model achieved this threshold for each feature selection method. In this data set and in the formulation of the problem, SBS still provides the best performance metric, as shown in Fig. 13. The results presented in Table 13 indicate that SBS achieves a score greater than 99% in all performance metrics.

7 features selected by the SBS and DT pair are as follows: {HH_L1_pcc, HH_L0.01_magnitude, HH_jit_L1_std, HH_jit_0.01_weight, HpHp_L1_pcc, HpHp_L0.01_weight, HpHp_L0.01_magnitude}. The distributions of the features according to the category of features and the duration of the time window are given in Fig. 14. When this feature set is compared to the selected feature sets in N-BaIoT, it is observed that the channel category is the dominant category instead of the host-based one. As MedBIoT covers malicious activities regarding the the C&C and formation phases of the botnet life cycle, the features that characterize host-to-host communications become more important. In contrast, N-BaIoT, which covers the attack phase, can discriminate malicious activities based on host-based features.

Similar to N-BaIoT, MedBIoT does not show any specific pattern on time periods, indicating whether longer or shorter periods are preferred. Although the longest period, 1 minute, provides more discriminative features among the others, still,

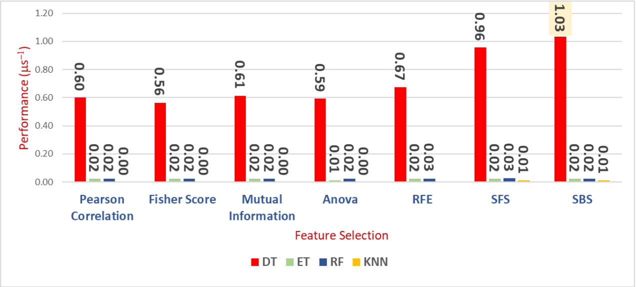
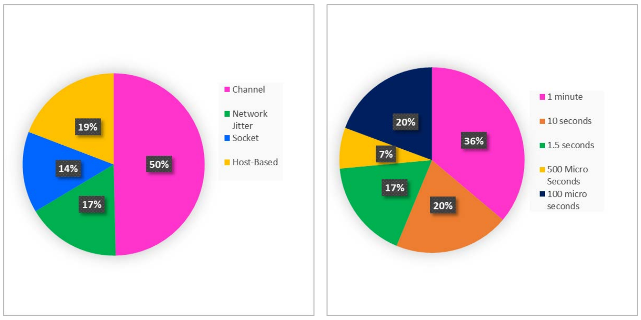


FIGURE 13. Performance achieved by binary classification models over the MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.



(a) Feature Category (b) Time Window

FIGURE 14. Feature category and time window contribution in each feature set for binary classification over the MedBIoT datase.

the second-best category is 100 microseconds, which is the smallest one.

2) 3-CLASS CLASSIFICATION

The 3-class classification of the MedBIoT dataset aims to identify whether the instance that represents a portion of network traffic belongs to the spreading or C&C phases of a botnet life cycle. The third class in this formulation is benign traffic. Similarly to binary classification, filter methods require a greater number of features to achieve high detection rates. More specifically, features 42, 38 and 49 should be included by Fisher score, Mutual information, and Anova, respectively, to achieve 98% detection rate (see Fig. 15. Wrapper methods, SFS and SBS, identified a set with 7 features. On the other hand, RFE requires 24-27 features.

SBS and DT are still the best pair of models and feature selection methods, as shown in Fig. 15. This highest performance is obtained from the following feature set: {HH_L3_magnitude, HH_L0.01_weight, HH_L0.01_radius, HH_jit_L1_weight, HH_jit_L0.1_std, HpHp_L5_pcc, HpHp_L0.1_magnitude}. The detection results given in Fig. 15 indicate that it is possible to find learning models for each feature selection method that gives a performance greater than 99%.

TABLE 14. Summary of the classification results with the selected model and feature sets based on the performance of the 3-class classification in the Med BIoT dataset.

Feature Selection	Class Name	Class Name	Accuracy	Precision	Recall	F1-Score
SBS	DT	Benign	0.99	0.98	0.99	0.99
		CC	0.99	0.99	0.99	0.99
		Spread	0.99	0.98	0.99	0.99

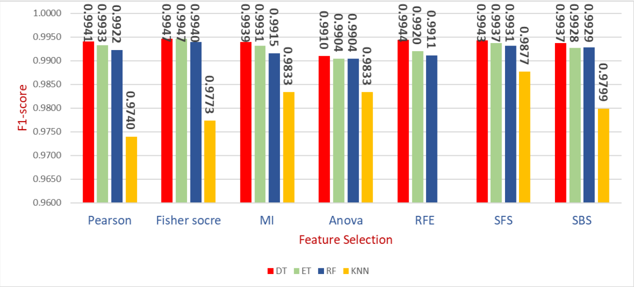


FIGURE 15. F1-scores for 3-class classification models in the MedBIoT dataset using feature subsets (see in Table 4&5) of feature selection algorithms.

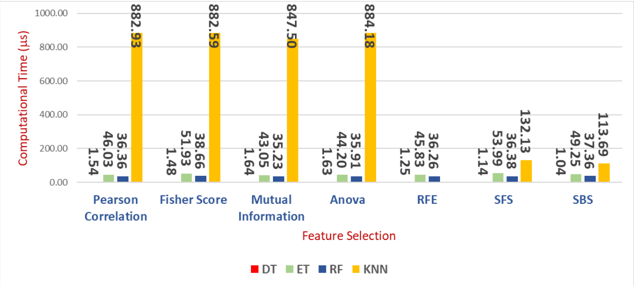


FIGURE 16. Computational Time required to classify a sample using 3-class classification models on MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

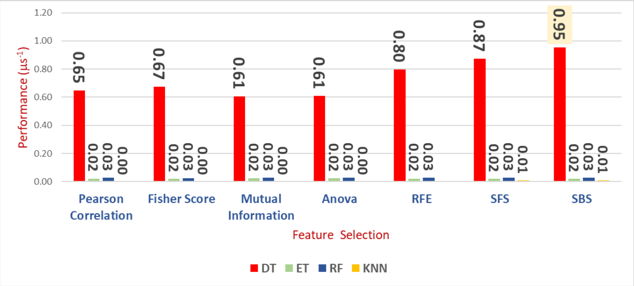
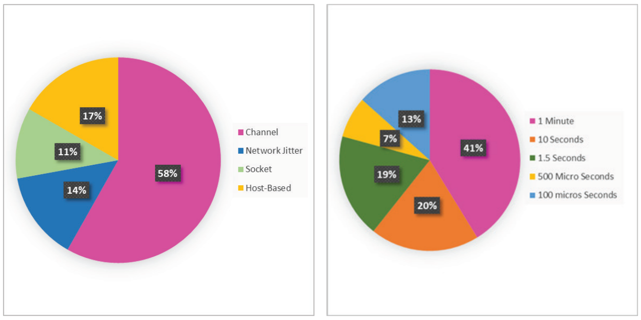


FIGURE 17. Performance achieved by 3-class classification models in the MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

Fig. 18 shows that channel-based features are more useful than other network categories to achieve the highest performance. Compared to binary classification, the ratios of channel features are more frequent. The time window results are similar to the binary classification outcome.

3) 4-CLASS CLASSIFICATION

In the 4-class classification, we consider the identification of the source malware that generates malicious traffic. Thus, the labels in this formulation are Mirai, Bash-Lite, Torii, and Benign. Fisher score, mutual information, and Anova require 46, 41 and 52 features, respectively



(a) Feature Category (b) Time Window

FIGURE 18. Feature category and time window contribution in each feature set for 3-class classification in MedBIoT dataset.

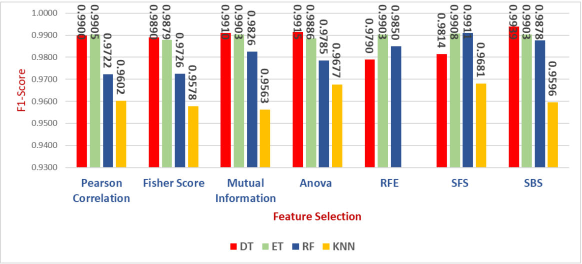


FIGURE 19. F1-scores for 4-class classification models in the MedBIoT dataset using feature subsets (see in Table 4&5) of feature selection algorithms.

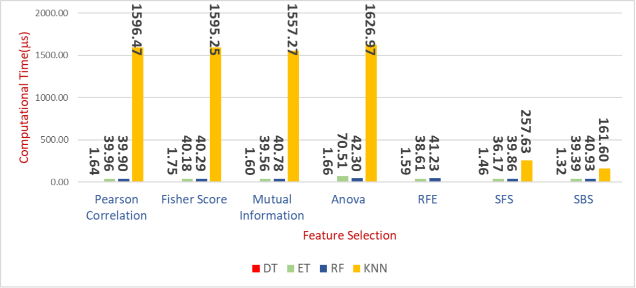


FIGURE 20. Computational Time required to classify a sample by 4-class classification models in the MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.

(see Table 4). SBS and SFS methods with any learning model achieve a higher detection with 7 features, whereas the feature numbers within the range of 22-29 are sufficient in RFE. F1 score of 99% can be achieved by a learning model in each feature selection method, as shown in Fig. 19. SBS and DT are the best pair of performers and use the following feature set:{MI_dir_L0.1_weight, HH_L1_pcc, HH_L0.01_magnitude, HH_jit_L0.01_weight, HH_jit_L0.01_std, HpHp_L0.01_weight, HpHp_L0.01_std}. Fig. 22 shows that the channel category is the most important category.

4) STANDARD FEATURE SET FOR BINARY, 3-CLASS AND 4-CLASS CLASSIFICATION TASKS OVER MedBIoT DATASET

To find a standard feature set that works for binary, 3-class and 4-class classification problems in the MedBIoT data set, similar to the case of N-BaIoT, we tested the performance

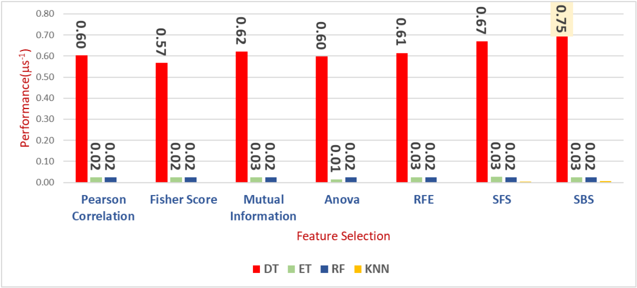
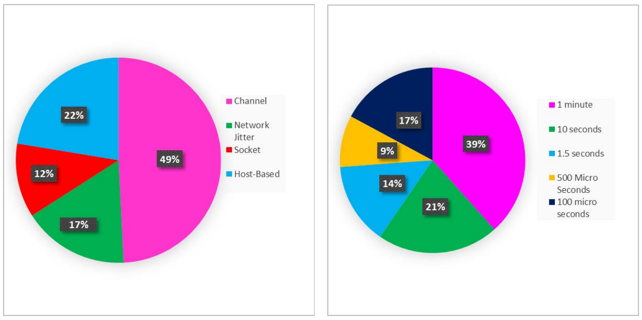


FIGURE 21. Performance achieved by 4-class classification models on the MedBIoT dataset using feature sets (see in Table 4 & 5) of feature selection methods.



(a) Feature Category (b) Time Window

FIGURE 22. Feature category and time window contribution in each feature set for 4-class classification in the MedBIoT dataset.

of the selected feature set of one classification on the other classification problem. We identified that the feature set of 4-class classification also works better in all other classifications, as shown in Table 15.

V. DISCUSSION

In this study, it is shown that all the machine learning problem formulations realized for the detection of IoT botnet attacks in two datasets, N-BaIoT and MedBIoT, achieved high detection performance in more than 99% with a limited number of features (i.e. 3 and 7 features).

In our experiments, we used various filter and wrapper methods for feature selection, in addition to four main machine learning methods to induce the models. In the case where we use filter methods, the results of feature selection are fed into the models. In wrapper methods, models are used directly for the assessment of feature subset alternatives. Performance evaluation was carried out based on the relationship between the F1 score and the computational time required to classify a sample. The wrapper method, SBS, with the DT model has achieved the most satisfactory trade-off between detection capacity and computational cost, exceeding the other alternative feature selection and learning model pairs.

Using feature selection approaches, tree-based models (DT, ET, and RF) achieved the best results in all classification types for both datasets, especially in multiclass classification types. k-NN classifier was not suitable for multiclass

TABLE 15. Summary of classification results using the standard 7-Feature Set for binary, 3-class and 4-class classification tasks in the MedBIoT dataset.

Classification Type	Class Name	Accuracy				Precision				Recall				F1-Score			
		DT	RF	KNN	ET	DT	RF	KNN	ET	DT	RF	KNN	ET	DT	RF	KNN	ET
Binary	Benign	0.99	0.99	0.96	0.99	0.99	0.99	0.95	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.96	0.99
	Attack	0.99	0.99	0.98	0.99	0.99	0.99	0.97	0.99	0.99	0.99	0.94	0.99	0.99	0.99	0.96	0.99
3-class	Benign	0.99	0.99	0.97	0.99	0.99	0.98	0.96	0.98	0.98	0.99	0.98	0.99	0.99	0.99	0.97	0.99
	CC	0.99	0.99	0.97	0.99	0.98	0.99	0.97	0.99	0.99	0.99	0.97	0.99	0.99	0.99	0.97	0.99
	Spread	0.99	0.98	0.97	0.99	0.99	0.99	0.98	0.99	0.99	0.98	0.96	0.99	0.99	0.98	0.97	0.99
4-class	Bashlite	0.99	0.99	0.94	0.99	1	1	0.97	0.99	0.99	0.99	0.94	0.98	0.99	0.99	0.96	0.99
	Benign	0.99	0.99	0.94	0.99	0.99	0.98	0.91	0.98	0.98	0.98	0.92	0.99	0.98	0.98	0.91	0.98
	Mirai	0.99	0.99	0.89	0.99	0.98	0.98	0.88	0.98	0.99	0.99	0.89	0.98	0.98	0.98	0.88	0.98
	Torii	0.99	0.99	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.98	0.99

classification and also took the longest computational time to classify the sample compared to tree-based models.

However, there are some differences between the results of the MedBIoT and N-BaIoT data sets. The former requires seven features, whereas three features in the latter data set are enough for high detection rates. Compared to N-BIoT, which addresses the attack stage of the botnet lifecycle, MedBIoT differentiates post-attack and C&C phases. It can be argued that the detection at the attack stage would be relatively easier, as this stage is usually accomplished by sending an enormous number of packets (i.e., spam, packet flooding). Therefore, more features are needed for other attack stages.

On the other hand, we observed a remarkable difference between filter and wrapper methods in some classification formulations. High accuracy rates are achieved with more than 28 features with filter methods for 9-class classification with N-BaIoT and all classifications with MedBIoT. On the other hand, the wrapper methods, SFS and SBS, identify an optimal set with 3 and 7 features for the respective formulations. One interesting observation is that the wrapper method, RFE, demonstrates quite different results for these formulations when compared to the other wrapper methods, so that, similarly to filter methods, it demands a high number of features. RFE applies a greedy approach by evaluating each feature one by one. Despite the differences in the statistical approach, filter methods also evaluate features in a similar fashion, one by one; thus, more composite feature set evaluation of SFS and SBS provides remarkable results in our case.

Another significant finding is obtained by comparing the feature categories that are prioritized by the feature selection methods. We identified that host-based features are more influential for the N-BaIoT dataset, whereas channel-based features show a more discriminatory property for the MedBIoT dataset. As the latter data set focuses on the spreading and C&C activity of the IoT malware within the target network, statistical features that are derived by tracking which network node communicates with which other node help more discriminate the malicious activity from the benign one or determine the type of malicious activity.

We conducted additional experiments to demonstrate the influence of feature categories. For this purpose, we induced models with only the features of the corresponding categories and reported the F1 scores for the ET, RF, DT and kNN models. As shown in Figure 23, the use of all host features achieves a perfect model with a 1.00 F score, while network jitter would be helpful for higher rates for the N-BaIoT data set. However, the features of the channel category achieve 99% rates, and the host and network jitter categories would also be helpful for MedBIoT, as demonstrated in Figure 24.

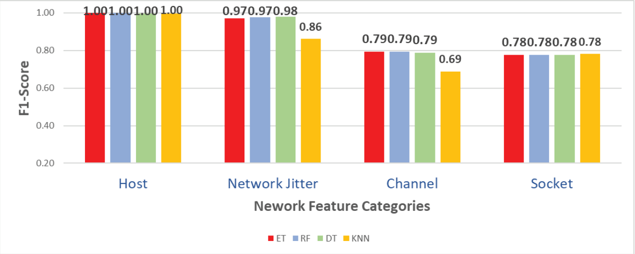


FIGURE 23. Comparison according to the feature categories - N-BaIoT dataset.

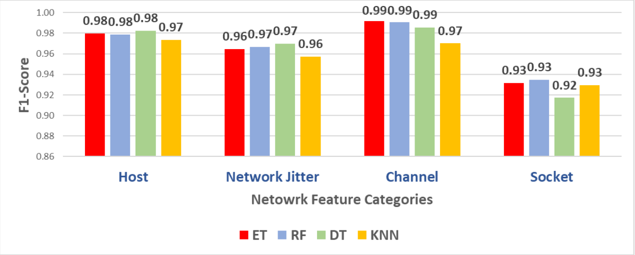


FIGURE 24. Comparison according to the feature categories - MedBIoT data set.

Our results send a significant message to experts who design intrusion detection systems. The attacks originating from the bots (i.e., as simulated in the N-BaIoT dataset) can be easily detected by the sensors that track the incoming and outgoing packet statistics without considering the destination of the traffic. However, post-attack and C&C stages require the sensors to follow the sources and targets of traffic flows. Although some feature selection methods utilize the features of the socket category, the overall picture shows that the identification of receiving parties would be enough without using the source and destination ports.

Our comparison regarding the feature categories from the time interval perspective shows that the longest interval value,

TABLE 16. Comparison of selected feature counts and classification results with previous work.

DataSet	Classification Type	Feature Selection method	Number of features	Model	Accuracy	Precision	Recall	F1-score
N-BaIoT	9-class	LOGISTIC REGRESSION METHOD [21]	19	ANN	96.4	93.9	95.1	99.13
		Random forest [55]	40	XGB	99.96	NA	NA	99.94
		correlation based feature selection (CFS) [56]	75	LSTM	97.84	97.81	95	96.25
		In this paper	**3**	**DT**	**99.57**	**99.56**	**99.55**	**99.55**
MedBIoT	Binary	Chi-Squared [57]	20	DT	95.3	95	98	96
				RF	95.3	95	98	96
		Original MedBIoT [23]	NA	DT	93.15	94.48	93.15	92.93
		In this paper	**7**	**DT**	**99.34**	**99.33**	**99.32**	**99.34**
	4-class	Original MedBIoT [23]	NA	DT	95.16	95.84	95.16	94.99
		In this paper	**7**	**DT**	**99.41**	**99.36**	**99.38**	**99.46**

*NA -Not Applicable.

1 min, contributes more to the set with a higher discrimination property.

We also compared our proposal with the latest methods from recent models, and the results are summarized in Table 16. For the N-BaIoT dataset, the 9-class classification achieved better results with lower subsets of features than others. Abbasi *et al.* provided the 19 most important features for various attacks using LR(logistic regression) as feature selection. With these features, the ANN model performed well with 96.4% accuracy, 93.9% precision, 93.9% recall, 99.13% F1 score [21]. Parra *et al.* has created an LSTM model with the help of correlation-based feature selection to classify attacks and confirmed its model with 75 features achieved 97.84% precision, 97.81% precision, 95% Recall, 96.25% F1 score [56]. Faysal *et al.* proposed an XGBoost model that used 40 related features and stated that the model classified attacks with 99.96% accuracy and 99.94% F1 score [55].

Compared to existing models, the proposed framework achieved for botnet attack type classification in the N-baIoT dataset achieved good detection performance on the SBS-DT model (decision tree) with three features: precision of 99.57%, precision of 99.56%, recall of 99.55% and F1 score of 99.55%. The proposed methodology effectively distinguishes IoT botnet attacks from network traffic with high detection rates.

However, feature selection is applied less on the MedBIoT dataset. Gandhi and Li has proposed the decision tree, random forest models for binary classification, and selection of chi-square characteristics utilized. Twenty features used to detect malware type for DT and RF models with 99.3% precision, 95% precision, 98% recall, 96% recall. We also compared our detection rates with an original MedBIoT dataset.

Our proposed methodology achieved good detection performance for binary and 4-class classification in the MedBIoT dataset compared to the SBS-DT models. For binary classification, 99.34% precision, 99.33% precision, 99.32% recall, 99.34% f1 score. For 4-class classification, 99.41% precision, 99.36% precision, 99.38% recall, 99.46% f1 score. To maximize the classification performance of the learning model, a random search is used to determine the best set of hyperparameters for each classification formulation and is summarized in Table 17.

TABLE 17. SBS-DT optimal parameters for each classification in the dataset.

Dataset	Classification Type	Description	Best Parameter
N-BaIoT	Binary	Maximum Depth of the tree	38
		Minimum number of samples required to split an internal node	15
		the minimum number of samples required to be at a leaf node.	1
		the impurity of a split	Entropy
	3-class	Maximum Depth of the tree	19
		Minimum number of samples required to split an internal node	3
		the minimum number of samples required to be at a leaf node.	3
		the impurity of a split	Gini
	9-class	Maximum Depth of the tree	24
		Minimum number of samples required to split an internal node	3
		the minimum number of samples required to be at a leaf node.	2
		the impurity of a split	Entropy
MedBIoT	Binary	Maximumn Depth of the tree	47
		Minimum number of samples required to split an internal node	26
		the minimum number of samples required to be at a leaf node.	8
		the impurity of a split	Entropy
	3-class	Maximum Depth of the tree	24
		Minimum number of samples required to split an internal node	15
		the minimum number of samples required to be at a leaf node.	3
		the impurity of a split	Entropy
	4-class	Maximumn Depth of the tree	39
		Minimum number of samples required to split an internal node	5
		the minimum number of samples required to be at a leaf node.	1
		the impurity of a split	Gini

VI. CONCLUSION

Botnet attacks change the shape and volume to deplete the target resources on the entire IoT network system. Therefore, to mitigate the critical impact, a machine learning-based intrusion detection system is developed to accurately classify botnet attacks.

In this work, we propose a reduced set of features to detect and classify malicious activities of popular IoT botnet malware. We identified six different binary or multiclass classification problems using datasets, N-BaIoT and MedBIoT. We applied various filter and wrapper methods with four machine learning methods to these datasets. Finally, we derive an optimal set of features for each classification problem. To our knowledge, no detailed comparison between the optimal feature sets required for different classification problems of IoT botnet detection, which can vary depending on the stage of the botnet life cycle, has been done before.

We obtained very high detection rates for each classification problem with fewer features. The decision tree-based SBS takes less time to classify the samples with the highest detection rate. Wrapper methods, SFS and SBS, were effective in finding the optimal feature sets in each classification.

Filter methods provide suboptimal results in terms of feature numbers for 9-class classification with N-BaIoT and all classifications with MedBIoT. Host-based features are more instrumental in the detection rates for N-BaIoT, whereas channel features play a more important role for MedBIoT.

REFERENCES

[1] P. Srinivasulu, M. S. Babu, R. Venkat, and K. Rajesh, "Cloud service oriented architecture (CSoA) for agriculture through Internet of Things (IoT) and big data," in *Proc. IEEE Int. Conf. Electr., Instrum. Commun. Eng. (ICEICE)*, Apr. 2017, pp. 1–6.

[2] R. Shah and A. Chircu, "IoT and AI in healthcare: A systematic literature review," *Issues Inf. Syst.*, vol. 19, no. 3, pp. 1–9, 2018.

[3] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.

[4] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.

[5] N. Hoque, D. K. Bhattacharyya, and J. K. Kalita, "Botnet in DDoS attacks: Trends and challenges," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2242–2270, 4th Quart., 2015.

[6] G. K. Kishore and K. Rajesh, "Avoiding attacks using node position verification in mobile ad hoc networks," in *Next-Generation Networks*. Singapore: Springer, 2018, pp. 111–118.

[7] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your botnet is my botnet: Analysis of a botnet takeover," in *Proc. 16th ACM Conf. Comput. Commun. Secur. (CCS)*, 2009, pp. 635–647.

[8] R. C. Perkins, C. J. Howell, C. E. Dodge, G. W. Burruss, and D. Maimon, "Malicious spam distribution: A routine activities approach," *Deviant Behav.*, vol. 43, no. 2, pp. 196–212, Feb. 2022.

[9] J. Leonard, S. Xu, and R. Sandhu, "A framework for understanding botnets," in *Proc. Int. Conf. Availability, Rel. Secur.*, 2009, pp. 917–922.

[10] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in Internet of Things," *J. Netw. Comput. Appl.*, vol. 84, pp. 25–37, Apr. 2017.

[11] M. Roesch, "Snort: Lightweight intrusion detection for networks," in *Proc. LISA*, vol. 99, Jun. 1999, pp. 229–238.

[12] E. Albin and N. C. Rowe, "A realistic experimental comparison of the Suricata and Snort intrusion-detection systems," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 122–127.

[13] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 2, pp. 1153–1176, 2nd Quart., 2016.

[14] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: A data perspective," *ACM Comput. Surveys*, vol. 50, no. 6, pp. 1–45, Nov. 2018, doi: 10.1145/3136625.

[15] J. M. H. Jimenez and K. Goseva-Popstojanova, "The effect on network flows-based features and training set size on malware detection," in *Proc. IEEE 17th Int. Symp. Netw. Comput. Appl. (NCA)*, Nov. 2018, pp. 1–9.

[16] H. Bahsi, S. Nomm, and F. B. La Torre, "Dimensionality reduction for machine learning based IoT botnet detection," in *Proc. 15th Int. Conf. Control, Autom., Robot. Vis. (ICARCV)*, Nov. 2018, pp. 1857–1862.

[17] E. B. Beigi, H. H. Jazi, N. Stakhanova, and A. A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Proc. IEEE Conf. Commun. Netw. Secur.*, Oct. 2014, pp. 247–255.

[18] F. V. Alejandre, N. C. Cortes, and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms," in *Proc. Int. Conf. Electron., Commun. Comput. (CONIELECOMP)*, 2017, pp. 1–7.

[19] A. Guerra-Manzanares, H. Bahsi, and S. Nomm, "Hybrid feature selection models for machine learning based botnet detection in IoT networks," in *Proc. Int. Conf. Cyberworlds (CW)*, Oct. 2019, pp. 324–327.

[20] F. Beer and U. Buhler, "Feature selection for flow-based intrusion detection using rough set theory," in *Proc. IEEE 14th Int. Conf. Netw., Sens. Control (ICNSC)*, May 2017, pp. 617–624.

[21] F. Abbasi, M. Naderan, and S. E. Alavi, "Anomaly detection in Internet of Things using feature selection and classification based on logistic regression and artificial neural network on N-BaIoT dataset," in *Proc. 5th Int. Conf. Internet Things Appl. (IoT)*, May 2021, pp. 1–7.

[22] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici, "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul./Sep. 2018.

[23] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network," in *Proc. 6th Int. Conf. Inf. Syst. Secur. Privacy*, 2020, pp. 207–218.

[24] G. Gu, J. Zhang, and W. Lee, "BotSniffer: Detecting botnet command and control channels in network traffic," in *Proc. 16th Annu. Netw. Distrib. Syst. Secur. Symp. (NDSS)*, San Diego, CA, USA, Feb. 2008, pp. 235–252.

[25] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "BotMiner: Clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. 17th Conf. Secur. Symp.*, CA, USA, 2008, pp. 139–154.

[26] M. S. Gadelrab, M. ElSheikh, M. A. Ghoneim, and M. Rashwan, "BotCap: Machine learning approach for botnet detection based on statistical features," *Int. J. Commun. Netw. Inf. Secur.*, vol. 10, no. 3, p. 563, Apr. 2022.

[27] M. Injadat, A. Moubayed, and A. Shami, "Detecting botnet attacks in IoT environments: An optimized machine learning approach," in *Proc. 32nd Int. Conf. Microelectron. (ICM)*, Dec. 2020, pp. 1–4.

[28] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[29] H.-T. Nguyen, Q.-D. Ngo, and V.-H. Le, "IoT botnet detection approach based on PSI graph and DGCNN classifier," in *Proc. IEEE Int. Conf. Inf. Commun. Signal Process. (ICICSP)*, Sep. 2018, pp. 118–122.

[30] Y. M. P. Pa, S. Suzuki, K. Yoshioka, T. Matsumoto, T. Kasama, and C. Rossow, "IoTPOT: A novel honeypot for revealing current IoT threats," *J. Inf. Process.*, vol. 24, no. 3, pp. 522–533, 2016.

[31] C. Yin, Y. Zhu, S. Liu, J. Fei, and H. Zhang, "An enhancing framework for botnet detection using generative adversarial networks," in *Proc. Int. Conf. Artif. Intell. Big Data (ICAIBD)*, May 2018, pp. 228–234.

[32] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Comput. Secur.*, vol. 31, no. 3, pp. 357–374, 2012.

[33] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, "Hybrid deep learning for botnet attack detection in the Internet-of-Things networks," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021.

[34] M. Alauthman, N. Aslam, M. Al-kasassbeh, S. Khan, A. Al-Qerem, and K.-K. R. Choo, "An efficient reinforcement learning-based botnet detection approach," *J. Netw. Comput. Appl.*, vol. 150, Jan. 2020, Art. no. 102479.

[35] B. Rahbarinia, R. Perdisci, A. Lanzi, and K. Li, "PeerRush: Mining for unwanted P2P traffic," in *Proc. Int. Conf. Detection Intrusions Malware, Vulnerability Assessment*. Berlin, Germany: Springer, 2013, pp. 62–82.

[36] K. Singh, S. C. Guntuku, A. Thakur, and C. Hota, "Big data analytics framework for peer-to-peer botnet detection using random forests," *Inf. Sci.*, vol. 278, pp. 488–497, Sep. 2014.

[37] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Jan. 2003.

[38] M. Alauthaman, N. Aslam, L. Zhang, R. Alasem, and M. A. Hossain, "A P2P botnet detection scheme based on decision tree and adaptive multilayer neural networks," *Neural Comput. Appl.*, vol. 29, no. 11, pp. 991–1004, 2018.

[39] S. Saad, I. Traore, A. Ghorbani, B. Sayed, D. Zhao, W. Lu, J. Felix, and P. Hakimian, "Detecting P2P botnets through network behavior analysis and machine learning," in *Proc. 9th Annu. Int. Conf. Privacy, Secur. Trust*, Jul. 2011, pp. 174–180.

[40] J. Velasco-Mata, V. Gonzalez-Castro, E. F. Fernandez, and E. Alegre, "Efficient detection of botnet traffic by features selection and decision trees," *IEEE Access*, vol. 9, pp. 120567–120579, 2021.

[41] S. García, M. Grill, J. Stiborek, and A. Zunino, "An empirical comparison of botnet detection methods," *Comput. Secur.*, vol. 45, pp. 100–123, Sep. 2014.

[42] P. Narang, J. M. Reddy, and C. Hota, "Feature selection for detection of peer-to-peer botnet traffic," in *Proc. 6th ACM India Comput. Conv.*, 2013, pp. 1–9.

[43] A. Pektaş and T. Acarman, "Effective feature selection for botnet detection based on network flow analysis," in *Proc. Int. Conf. Automatics Informat.*, 2017, pp. 1–4.

[44] C. Aggarwal, *Data Mining*. Cham, Switzerland: Springer, 2015.

[45] Q. Gu, Z. Li, and J. Han, "Generalized Fisher score for feature selection," 2012, *arXiv:1202.3725*.

[46] P. A. Estévez, M. Tesmer, C. A. Perez, and J. M. Zurada, ''Normalized mutual information feature selection,'' *IEEE Trans. Neural Netw.*, vol. 20, no. 2, pp. 189–201, Feb. 2009.

[47] S. Beniwal and J. Arora, ''Classification and feature selection techniques in data mining,'' *Int. J. Eng. Res. Technol.*, vol. 1, no. 6, pp. 1–6, 2012.

[48] M. ShakilPervez and D. Md. Farid, ''Literature review of feature selection for mining tasks,'' *Int. J. Comput. Appl.*, vol. 116, no. 21, pp. 30–33, Apr. 2015.

[49] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik, ''Gene selection for cancer classification using support vector machines,'' *Mach. Learn.*, vol. 46, nos. 1–3, pp. 389–422, 2002.

[50] M. Kudo and J. Sklansky, ''Comparison of algorithms that select features for pattern classifiers,'' *Pattern Recognit.*, vol. 33, no. 1, pp. 25–41, 2000.

[51] Y. Sasaki, ''The truth of the F-measure,'' Univ. Manchester, Manchester, U.K., Lect. Notes, Oct. 2007. [Online]. Available: https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf

[52] A. McCallum and K. Nigam, ''A comparison of event models for naive Bayes text classification,'' in *Proc. AAAI Workshop Learn. Text Categorization*, 1998, vol. 752, no. 1, pp. 41–48.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, and M. Blondel, ''Scikit-learn: Machine learning in Python,'' *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Jan. 2011.

[54] S. Raschka, ''MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack,'' *J. Open Source Softw.*, vol. 3, no. 24, p. 638, Apr. 2018.

[55] J. A. Faysal, S. T. Mostafa, J. S. Tamanna, K. M. Mumenin, M. M. Arifin, M. A. Awal, A. Shome, and S. S. Mostafa, ''XGB-RF: A hybrid machine learning approach for IoT intrusion detection,'' *Telecom*, vol. 3, no. 1, pp. 52–69, Jan. 2022.

[56] G. De La Torre Parra, P. Rad, K.-K.-R. Choo, and N. Beebe, ''Detecting Internet of Things attacks using distributed deep learning,'' *J. Netw. Comput. Appl.*, vol. 163, Aug. 2020, Art. no. 102662.

[57] R. Gandhi and Y. Li, ''Comparing machine learning and deep learning for IoT botnet detection,'' in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, Aug. 2021, pp. 234–239.

RAJESH KALAKOTI (Graduate Student Member, IEEE) received the master's degree in information technology from JNTUK, Kakinada. He is currently pursuing the Ph.D. degree with the Tallinn University of Technology. He worked as a Lecturer of information technology and a Software Engineer in India. He is also an Early Stage Researcher of XAI in cyber security domain with the Tallinn University of Technology. He has (co)authored three publications in conference proceedings. His research interests include the IoT security, network traffic, malware, HTTP-based botnet, machine learning, and XAI. He is a member of the IEEE Computer Society.

SVEN NÕMM received the Ph.D. degree jointly from the Tallinn University of Technology (TalTech), Estonia, and the École Centrale de Nantes et L'Université de Nantes, France, in 2004. He is currently a Senior Research Scientist at the Department of Software Science, TalTech. He has published more than 100 articles in scientific. His research interests include human–machine interaction, analysis of human motions, and applications of AI to the problems of cybersecurity and geoscience.

HAYRETDIN BAHSI received the M.Sc. degree in computer engineering from Bilkent University and the Ph.D. degree in computer engineering from Sabanci University. He is currently a Research Professor at the Center for Digital Forensics and Cyber Security, Tallinn University of Technology, Tallinn, Estonia. His research interests include cyberphysical system security and the application of machine learning methods to cybersecurity problems.