





## Optimizing Multi-Class Botnet Detection Models to Detect SPAM Botnets with Feature Selection Methods: A Comparative Analysis



Java Kanaya Prada<sup>1</sup>, Tohari Ahmad<sup>1\*</sup>, Dandy Pramana Hostiadi<sup>2</sup>, Muhammad Aidiel Rachman Putra<sup>1</sup>

<sup>1</sup> Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

<sup>2</sup> Department of Magister Information Systems, Institut Teknologi dan Bisnis STIKOM Bali, Bali 80234, Indonesia

Corresponding Author Email: [tohari@its.ac.id](mailto:tohari@its.ac.id)

Copyright: ©2024 The authors. This article is published by IETA and is licensed under the CC BY 4.0 license (<http://creativecommons.org/licenses/by/4.0/>).

<https://doi.org/10.18280/ijse.140430>

### ABSTRACT

**Received:** 4 July 2024

**Revised:** 17 August 2024

**Accepted:** 26 August 2024

**Available online:** 30 August 2024

#### Keywords:

*botnet detection, spam, feature selection, network security, machine learning*

A botnet involves the use of illegal software to carry out malicious activities that pose a threat to network security, particularly through spam attack activities. Many studies have focused on developing detection models to categorize network activities as either botnet or non-botnet. However, there is still a need for research to identify spam activities within botnet activities, including improving the feature selection process. The aim of this research is to identify feature selection methods that can improve machine learning models for botnet detection, particularly in SPAM botnet detection. To address this, our research implements feature selection methods as a preprocessing step before classifying network activity data using a decision tree algorithm for botnet spam detection with multi-class classification. Feature selection during the data preprocessing phase is crucial, as it has been shown to enhance the performance of detection models. In this study, eight types of feature selection methods were implemented, yielding mixed results. Experimental findings indicate that the classification method using a decision tree without feature selection produced the best overall results, achieving a macro average F1-Score of 91.18%, a weighted average Precision of 99.07%, a Recall of 99.03%, an F1-Score of 99.05%, and an Accuracy rate of 99.03%. SelectKBest with chi2 Feature Selection slightly outperformed other methods in detecting SPAM Botnets, with a Recall of 87.93% and an F1-Score of 79.68%.

## 1. INTRODUCTION

Technology development in the cyber era impacts data and information security risks. One of the threats that often occurs is attacks involving malicious software known as Malicious Software (Malware). It is a program designed to disrupt, damage, or gain unauthorized access to computer systems or computing devices [1, 2]. One type of malware that often threatens information systems on the internet is a robot network called a botnet [3, 4]. Botnets refer to a collection of compromised computers, known as zombies or bots, operated under the command and control of a central server called bot master [5-7]. These botnets pose diverse threats, capable of executing actions ranging from Distributed Denial of Service (DDoS) attacks, click fraud, information theft, espionage, phishing, port scanning, and spam dissemination [4, 5]. Among these, spam is the most massive [8], causes the most damage, and prolific [9]. The elevated prevalence of spam threats can be attributed to its frequent use as an initial tactic for various purposes, including phishing and the infection of new devices to establish botnet networks [10].

This study emphasizes the urgent requirement for a robust system capable of effectively identifying botnet activities, particularly those associated with spam operations. While various botnet detection models utilizing methodologies like

extreme learning machine [1], machine learning [11, 12], deep learning [4, 13], clustering [14, 15], and hybrid analysis [16] already exist, the quest for an optimal model for classifying botnet-driven spam continues. Developing a detection model to identify botnet SPAM activities within network traffic poses a significant challenge. Botnet detection datasets often suffer from class imbalance, where the number of legitimate network traffic samples dramatically exceeds the number of botnet samples. This imbalance can lead to biased models that are less effective in detecting botnets [17]. Botnet detection involves identifying relevant features from network traffic that can accurately distinguish between botnet and non-botnet traffic. This is particularly challenging due to the data's correlated and mutually informed features [12, 18]. The reliability of such a detection model hinges on its ability to accurately categorize network traffic into three classes: normal, botnet non-spam, and botnet SPAM.

This research proposes a novel model for detecting botnet SPAM activities in response to this challenge. The model analyzes network traffic data, employing the decision tree algorithm in Machine Learning. Notably, the paper introduces innovations in data preprocessing, explicitly focusing on feature selection. The research systematically compares various feature selection methods within Machine Learning to identify the most suitable approach for detecting botnet-driven

spam. The goal of this experiment is to determine the impact of different feature selection methods on the process of creating a Machine Learning model for botnet detection. Various feature selection methods are tested on the data, and their effectiveness in detecting botnets, particularly botnet SPAM, is compared. The novelty of this paper lies in the comparison of diverse feature selection methods, especially for botnet SPAM detection. Unlike previous research, which typically focuses on binary classification to determine whether a network is a botnet or benign, this study delves deeper into the specific challenge of detecting botnet SPAM activities. The proposed detection model can mitigate the prevalence of botnet SPAM activities, preventing their spread and infection of new devices.

This paper comprises five sections. Section II discusses related research on botnet detection and feature selection. Section III offers a detailed presentation of the proposed approach. Subsequently, Section IV discusses the results and their evaluation. Finally, a conclusive summary is presented in Section V, wrapping up the key findings and insights discussed in the preceding sections.

## 2. RELATED WORKS

Previous studies have presented models for detecting botnet activity employing diverse methods, including extreme learning machine [1], machine learning [11, 12], deep learning [4, 13], clustering [14, 15], and hybrid analysis [16]. While developing multi-label botnet classifiers poses a significant challenge, research in the realm of multi-label classifiers within Intrusion Detection Systems (IDS) has experienced substantial growth. Ali et al. [19] highlighted that machine learning techniques are highly effective for botnet detection due to their accuracy. This justifies our research's focus on machine learning for botnet SPAM detection.

Several studies in botnet detection have explored diverse machine learning models to enhance the efficacy of identifying and mitigating botnet threats. The study of Alshamkhany et al. [11] utilized the Bot-IoT and University of New South Wales (UNSW) datasets to compare the performance of four popular classifiers: Naïve Bayes,  $k$ -Nearest Neighbor, Support Vector Machine, and Decision Trees. Among these models, the decision trees-based approach emerged as the most effective for botnet detection. The experiments, conducted with an extensive dataset comprising 82,000 records from the UNSW-NB15 dataset, demonstrated remarkable results. The decision trees model exhibited superior performance with an outstanding 99.89% testing accuracy, 100% precision, 100% recall, and 100% F-score in identifying and classifying botnet attacks. This outcome suggests that decision trees are a robust and reliable choice for botnet detection among the models investigated. Proved that decision trees have the potential to contribute significantly to advancing cybersecurity. Consequently, recognizing the effectiveness of the decision tree algorithm, it is employed as the chosen approach in this research, underscoring its relevance and applicability in the context of botnet detection.

Previous studies have consistently highlighted that feature selection is a crucial step in the preprocessing phase of machine learning, which has been proven to enhance the accuracy and efficiency of machine learning models [20, 21]. The selection of appropriate feature selection methods results in the identification of interrelated features, contributing to

improved performance in Machine Learning models.

Studies have reinforced the importance and effectiveness of feature selection in enhancing botnet detection. For instance, Al Tawil and Sabri [22] demonstrated that using the Moth Flame Optimization (MFO) technique reduced the number of features from 78 to 4, leading to a higher detection rate of 100% and an accuracy rate of 99.9% when employing Decision Trees as a classifier. Another study of Safitri et al. [23] found that selecting a certain number of features improved detection accuracy, achieving an average detection accuracy of 98.34% using four features, compared to 97.46% with 11 features. This underscores the significance of selecting the optimal number and type of features to boost the performance of botnet detection models. However, these studies primarily focused on binary classification rather than multi-class classification, which includes normal, botnet, and botnet SPAM detection.

Research of Kalakoti et al. [24] delves into various feature selection methods, testing binary classification and multi-class scenarios involving three and even nine classes. The study found that techniques such as sequential backward selection (SBS) and sequential forward selection (SFS) combined with decision trees were the most effective. Despite this comprehensive analysis, botnet SPAM detection still needs to be explicitly addressed.

Moreover, Lefoane et al. [25] explored botnet detection using feature selection and found that for Decision Trees, the results were consistent between Feature Selection (WFS) and No Feature Selection (NFS) methods, indicating that while feature selection might not always enhance accuracy, it can improve processing speed. This is particularly relevant to our study, where NFS emerged as the best approach when combined with Decision Trees. Consequently, this experiment also tests the condition without Feature Selection to evaluate its impact on botnet SPAM detection.

Prior research leveraging feature selection has demonstrated intriguing findings in the context of botnet detection. Analysis of network traffic features for botnet detection and feature selection using various methods revealed contradictory outcomes. This suggests that no feature is inherently detrimental for elimination in the context of botnet detection models [26]. This observation challenges conventional assumptions and explores whether the same holds for botnet SPAM detection.

Several previous studies introduced botnet detection models and showed high-performance detection. However, the problem of botnet detection with multi-class labels, especially SPAM, still needs to be widely addressed. This paper proposes an improvement for botnet SPAM multi-class detection, aiming to verify whether discarding features from the initial dataset is unnecessary for effective botnet SPAM detection, thereby shedding light on potential deviations from established practices in the field.

## 3. PROPOSED METHOD

This paper aims to enhance the performance of the botnet SPAM detection classification model by focusing on the data preprocessing method, namely feature selection. The proposed method consists of four main steps: data preprocessing, data labeling, data splitting, and the classification process using the decision tree method. The proposed method is also illustrated in Figure 1, visually representing the workflow.

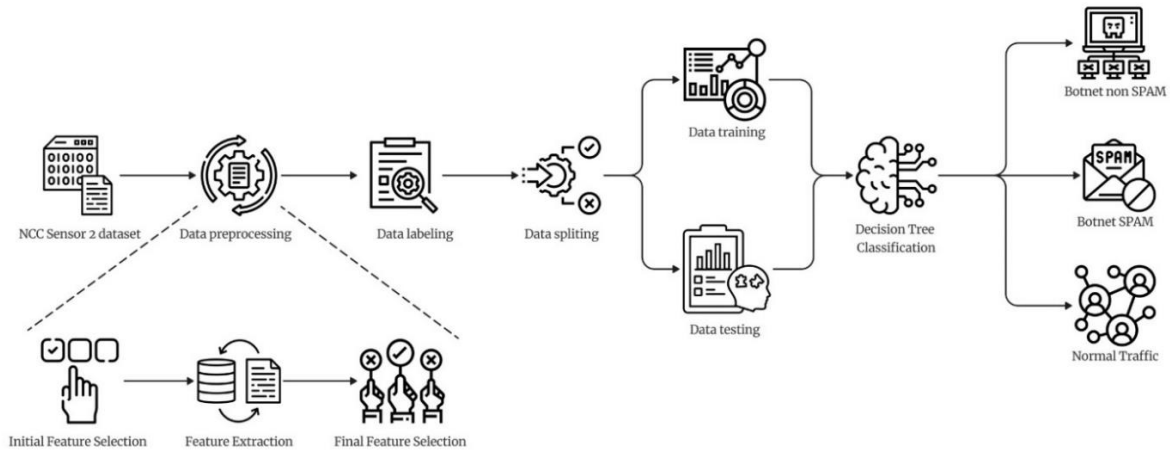


Figure 1. Proposed method

### 3.1 Data preprocessing

The data used in this study is a collection of network traffic data in binary network flow (.binetflow) format. The network traffic data has 18 features, namely StartTime, Dur, Proto, SrcAddr, Sport, Dir, DstAddr, Dport, State, sTos, dTos, TotPkts, TotBytes, SrcBytes, Label, ActivityLabel, BotnetName, and SensorId. Of these 18 features, only eight numerical and ten categorical features exist. These categorical features require preprocessing to facilitate the understanding of machine learning models. In this study, the data preprocessing process is divided into two main stages. The first stage is initial feature selection, followed by feature extraction from categorical features, and the final stage is the last feature selection using several methods.

#### 3.1.1 Initial feature selection

Referenced from Putra et al. [27], two numeric features, namely 'dTos' (7.74%) and 'sTos' (0.77%), with high null values, are eliminated. Additionally, the 'ActivityLabel' and 'SensorId' features are removed as they serve as labels. 'ActivityLabel' indicates whether network traffic is an attack, while 'SensorId' indicates which network traffic was recorded on which sensor.

#### 3.1.2 Feature extraction

Categorical features such as 'Proto', 'Dir', and 'State' undergo binary encoding to enhance their representation in machine learning models. Binary encoding is a technique that represents categorical data using binary code, converting each category into a unique binary sequence. After binary encoding, the dataset concludes the preparation phase with an expanded set of numerical features, enhancing the representation of categorical information in 'Proto,' 'Dir,' and 'State.' After the encoding, the current feature now is 21 features, excluding the target label.

#### 3.1.3 Final feature selection

A series of feature selection techniques were applied to identify the optimal subset for model training. Eight distinct methods, including No Feature Selection (NFS), SelectKBest: chi2 (SKB-C2), SelectKBest: ANOVA F-value (SKB-AF), SelectKBest: mutual\_info\_classif (SKB-MI), VarianceThreshold (VT), Backward Elimination (BE), Recursive Feature Elimination (RFE), and SelectFromModel: Tree-Based Feature Selection (SFM-TB), were employed. The outcome of each method, resulting in varying numbers of final

features, was thoroughly analyzed to ascertain the most effective feature set for subsequent classification.

The first method, No Feature Selection (NFS), serves as a baseline by retaining all features from the preprocessing step without applying any selection process. This method helps establish a benchmark for evaluating the effectiveness of feature selection.

SelectKBest (SKB) methods were employed in three variations, each using a different scoring function to select the top K features. SKB-C2 utilizes the Chi-Squared test, which measures the dependence between each feature and the target variable. This method effectively filters out features that are likely independent of the class and, therefore, are irrelevant for classification. SKB-AF uses the ANOVA F-value as its scoring function, selecting features that show the most significant linear relationship with the target variable by comparing the variance between groups to the variance within groups. SKB-MI relies on mutual information to estimate the dependency between features and the target variable, with higher mutual information values indicating more substantial dependencies, thus making the features more relevant for the model.

Another approach used was VarianceThreshold (VT), a simple baseline method that removes features with variance below a specified threshold. By default, it eliminates features with zero variance, which do not contribute predictive power to the model since they exhibit the same value across all samples.

Backward Elimination (BE) was also applied, starting with all features and iteratively removing the least significant ones based on p-values from statistical tests. This method continues until the optimal subset of features is identified, striking a balance between model performance and the number of features.

Recursive Feature Elimination (RFE) is another technique that was explored, which works by recursively removing the least essential features according to the weights assigned by an external estimator (e.g., coefficients in a linear model). The process is repeated until the desired number of features remains. When combined with cross-validation (RFECV), RFE can automatically determine the optimal number of features by assessing model performance across different cross-validation splits.

Finally, SelectFromModel (SFM-TB) was used for tree-based feature selection. This method leverages the feature importance scores generated by tree-based models, such as decision trees or random forests. These scores are based on the

reduction in impurity (e.g., Gini impurity or entropy), and features with low importance are discarded, leaving only those that significantly contribute to the model's predictions.

### 3.2 Data labeling

In the dataset, each network activity has its own information in the label feature; for this experiment, the target class will become three classes: normal, botnet non-SPAM, and botnet SPAM. Those classes are extracted from the label feature. That feature was systematically analyzed, with each word parsed to categorize network traffic into three classes: 'normal,' 'botnet non-SPAM,' and 'botnet SPAM.' Instances containing both "botnet" and "spam" were designated as 'Class 2 (botnet SPAM),' while those with only "botnet" were categorized as 'Class 1 (botnet non-SPAM).' Instances without reference to "botnet" were labeled as 'Class 0 (normal).'

### 3.3 Data splitting

Data splitting divides the dataset into two parts: one for training and the other for testing. This division follows a proportion of 70% for training and 30% for testing. The dataset being used is facing a significant imbalance, where the percentage of normal traffic is 92.43%, the percentage of the botnet non-SPAM is 6.97%, and for the botnet SPAM, 0.59%.

Due to the imbalanced nature of the botnet dataset, where normal network traffic dominates, a random splitting system cannot be directly employed. Each data resulting from the splitting must have a balanced proportion for each class. This is crucial to ensure the performance of the machine learning classification model.

The initial step in data splitting involves identifying network traffic for each class. Subsequently, each class is randomly divided into 70% and 30% ratios for training and testing. Finally, the three classes are merged into a new dataset for both training and testing. This method guarantees a balanced percentage of normal network traffic, non-botnet SPAM, and botnet SPAM for training and testing, addressing the inherent imbalance in the botnet dataset.

### 3.4 Classification

Classification is used to identify whether traffic is normal, botnet non-SPAM, or botnet SPAM. The classification method employed here is the decision tree algorithm. The decision tree is a classification method that model decisions in a tree structure. This decision tree consists of nodes representing decisions, branches representing the outcomes of these decisions, and leaves representing class labels. The decision tree algorithm selects the best features at each node based on specific criteria.

In this classification stage, after the data has undergone preprocessing, labeling, and splitting, the decision tree model is used to identify the category of each data instance. The decision tree will make decisions based on the features that have been selected and extracted previously.

## 4. RESULT AND DISCUSSION

This chapter analyzes the results and performance of botnet SPAM and non-SPAM detection using feature selection as a preprocessing step. The first subchapter discusses the details of the dataset used, the NCC-2 dataset [28], to test the

proposed model. The second subchapter analyzes the results of data preprocessing using feature selection before classification is performed. Lastly, the detection model's performance is analyzed in the third subchapter by comparing the score values.

### 4.1 Dataset

This experiment utilizes the NCC-2 [28] due to several factors: publicly available, quite popular [7, 29-31], and diverse botnet attack activities. This dataset recorded simultaneous botnet attack activities on three different sensors. This research explicitly uses NCC-2 sensor 3 to test the proposed method. The NCC-2 dataset with sensor 3 reveals various types of attacks conducted by different botnet entities over an 8-hour period: the Rbot, Neris, Murlo, NSIS.ay, and Virut botnet. The Rbot botnet engaged in IRC, PS, DDoS, and US attacks, while the Neris botnet participated in IRC, SPAM, CF, and PS scenarios. The Murlo botnet was associated with PS attacks, NSIS.ay with P2P scenarios, and Virut with SPAM, PS, and HTTP source scenarios.

The dataset consists of eight numerical features (duration of activities, source and destination type of service, total transactions in packets, overall transaction size, transaction size from source to destination, and sensor identity recording network activities) and ten categorical features (start time timestamp, source and destination ports, the protocol used in transactions, source and destination IP addresses, communication direction, data stream status, network traffic labels, activity labels, and botnet names). Combining these features, the dataset provides a comprehensive insight into various aspects of recorded network activities.

### 4.2 Feature selection result

From the 18 features in the dataset, encoding was done to obtain numerical values from 3 categorical feature columns with the least number of categories, namely 'Proto', 'Dir', and 'State'. Proto refers to the protocol used in the transaction (TCP, UDP, etc.), Dir represents the direction of communication carried out (->, <-, <->, etc.), and State provides information about the status of the data stream (SRPA\_SPA, FSPA\_FSPA, etc.). Since the State feature has 299 different categories, binary encoding was used.

**Table 1.** Feature selection result

Methods	Initial Feature	Final Feature
NFS		21
SKB-C2		10
SKB-AF		10
SKB-MI		10
VT	21	8
BE		9
RFE		10
SFM-TB		4

The encoding result transforms the dataset into 32 features. Labeling is then performed, and numerical features are selected, resulting in a final set of 21 numerical features. These 21 numerical features will then undergo feature selection.

Eight feature selection methods were applied: NFS, SKB-C2, SKB-AF, SKB-MI, VT, BE, RFE, and SFM-TB. Table 1 details the outcomes of each feature selection process conducted with these methods. Notably, the SFM-TB method yielded the smallest features after the selection process.

A key observation is that certain features, such as 'Dur' (Duration), 'TotBytes' (Total Bytes), and 'SrcBytes' (Source Bytes), were consistently selected by 7 out of the 8 feature selection methods. This consistency may be attributed to the continuous nature of these features, in contrast to the Boolean values generated from the earlier binary encoding of categorical features. Continuous features often provide richer information, making them more likely to be retained during the feature selection.

Finally, the selected features were used to perform classification using the decision tree method to evaluate the impact of different feature selection methods on the model's performance.

### 4.3 Result analysis

The experiment was conducted to test preprocessed data with a machine learning algorithm, Decision Tree. This experiment evaluates the model with four evaluation matrices: Accuracy, Precision (Prec.), Recall (Rec.), and F1-score (F1). In addition, there are other evaluation parameters, namely macro average and weight average.

As demonstrated in Table 2, all feature selection methods could detect regular traffic with very good Prec., Rec., and F1, with an average above 99.00%. Similarly, for detecting Botnet Non-SPAM, except for the feature selection method with SKB-AF, this method produced the lowest score compared to the other seven methods.

In the detection of Botnet SPAM, six methods produced similar scores, namely 72.00% for Prec., 87.90% for Rec., and 79.50% for F1. Besides, the other two methods produced more significant score differences. Especially in the SKB-AF, it scored the lowest, with 62.66% for Prec., 2.14% for Rec., and 4.14% for F1. The SKB-AF also shows the lowest score for the Macro Average, Weighted Average, and accuracy, as shown in Table 3.

The feature selection method SFM-TB produced the highest Prec. The score for Botnet SPAM is 84.65%, but this is not balanced with the Rec. score and F1. Rec. score and F1 for this

method yield lower values compared to the other six methods, namely 65.79% for Rec. and 74.04% for F1.

The higher Precision score of the SFM-TB method in detecting Botnet SPAM is likely due to the distinct set of features it selected, which differed significantly from those chosen by other methods. SFM-TB selected only four features: 'Dur', 'TotPkts', 'TotBytes', and 'SrcBytes', all continuous features. Notably, no binary features were selected by this method. Notably, the Botnet SPAM data was highly imbalanced, which may have influenced the classification results. Overall, it can be concluded that feature selection can improve the performance of minor class detection (in this study, Botnet SPAM). With SFM-TB, the Prec. value increases by 11.82%, while the Rec. value for the Botnet SPAM class is 0.02% superior to the feature selection results with BE and RFE, and the F1 value is 0.02% superior to the SKB-C2 feature selection. However, no feature selection method produces stable performance. Future work requires deeper analysis to combine multiple feature selection methods to obtain stable performance.

### 4.4 Comparative analysis

Compared with previous research which used two-stack decision tree algorithm with max-depth=12 [27], the two best-performing methods, NFS and SFM-TB, show better average results in Prec., Rec., F1, and Accuracy metrics. However, SFM-TB shows lower scores on the Rec. and F1 metrics for Botnet SPAM, its Prec. score was the highest.

For Accuracy, NFS is the highest scenario. The NFS method, which involves no feature selection, still performed well due to the more extensive feature extraction conducted in this experiment compared to the study of Putra et al. [27]. In contrast to the preprocessing method employed in the study of Putra et al. [27], our experiment introduced an essential modification. While previous research [27] encoded the "Proto" and "Dir" features, our approach extended this to include the State feature, thereby enhancing the comprehensiveness of our data processing.

**Table 2.** Comparison of detection performance – class

Method	Normal			Botnet Non-SPAM			Botnet SPAM		
	Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)
NFS	99.63	99.42	99.52	93.92	94.80	94.36	72.83	87.91	79.66
SKB-C2	99.63	99.41	99.52	93.75	94.80	94.27	72.84	87.93	79.68
SKB-AF	94.16	99.31	96.67	75.22	27.08	39.82	62.66	2.14	4.14
SKB-MI	99.62	99.42	99.52	93.91	94.75	94.33	72.82	87.91	79.66
VT	99.62	99.41	99.52	93.75	94.77	94.25	72.58	87.91	79.51
BE	99.62	99.41	99.52	93.78	94.77	94.27	72.82	87.93	79.67
RFE	99.63	99.41	99.52	93.79	94.79	94.28	72.82	87.93	79.67
SFM-TB	99.41	99.42	99.42	92.83	94.37	93.59	84.65	65.79	74.04
Putra et al. [27]	98.15	98.92	98.54	86.21	75.14	80.29	66.31	85.66	74.75

**Table 3.** Comparison of detection performance – average

Method	Macro Average			Weighted Average			Accuracy (%)
	Prec. (%)	Rec. (%)	F1 (%)	Prec. (%)	Rec. (%)	F1 (%)	
NFS	88.79	94.04	91.18	99.07	99.03	99.05	99.03
SKB-C2	88.74	94.05	91.16	99.06	99.02	99.03	99.02
SKB-AF	77.35	42.84	46.88	92.64	93.67	92.13	93.67
SKB-MI	88.78	94.03	91.17	99.06	99.03	99.04	99.03
VT	88.65	94.03	91.09	99.05	99.01	99.03	99.01
BE	88.74	94.04	91.15	99.06	99.02	99.03	99.02
RFE	88.75	94.04	91.16	99.06	99.02	99.04	99.02
SFM-TB	92.30	86.53	89.02	98.86	98.87	98.86	98.87
Putra et al. [27]	-	-	-	97.13	97.19	97.12	97.19

## 5. CONCLUSION

This research aims to create a model that can detect three classes in Botnet SPAM. The proposed model consists of 2 main parts: Data Preprocessing and Classification. The preprocessing compares feature selection methods, including eight types: No Feature Selection, SelectKBest: chi2, SelectKBest: ANOVA F-value, SelectKBest: mutual\_info\_classif, VarianceThreshold, Backward Elimination, Recursive Feature Elimination, and SelectFromModel: Tree-Based Feature Selection. The next part, classification, uses a Decision Tree.

Experiment results using the NCC-2 Sensor 3 dataset show that the Decision Tree classification method without feature selection produces the best performance, with a Macro Average F1-Score of 91.18%, Weighted Average Precision of 99.07%, Recall of 99.03%, F1-Score of 99.05%, and Accuracy of 99.03%. However, for Botnet SPAM detection, the SelectKBest with chi2 feature selection method performs 87.93% for Recall and 79.68% for F1-Score. In general, the proposed method performs better than previous studies that used two-stack decision tree algorithm with max-depth=12. Although weak in Precision, Recall, and F1-score metrics for the Botnet SPAM class, the proposed method excels in the Normal, Botnet Non-SPAM, Weighted Average, and Accuracy classes.

Future work will involve testing the ensemble feature selection methods with a broader range of machine learning classification algorithms, such as *k*-Nearest Neighbors (*k*-NN) and ensemble methods like XGBoost and AdaBoost. These algorithms offer different strengths in handling the complexity and variability of botnet traffic data, potentially leading to enhanced detection performance. Additionally, exploring advanced feature extraction techniques is crucial, given the unique characteristics of network traffic data. Traditional methods like one-hot encoding are impractical due to the high dimensionality they introduce, so future research will focus on more sophisticated feature extraction approaches, such as embeddings or unsupervised learning techniques like autoencoders, to capture the nuances of network traffic more effectively.

Moreover, hyperparameter tuning will be a crucial area of exploration, not just for the classification algorithms but also for the feature selection methods. Fine-tuning these parameters could significantly improve model performance and generalization capabilities. Another avenue for future research is integrating deep learning approaches, which may offer more robust performance on complex datasets. Finally, real-world testing and validation of these models in live network environments will be essential to assess their practical applicability and reliability in detecting botnet activity under diverse and evolving network conditions.

## ACKNOWLEDGMENT

This work has been supported by the Institut Teknologi Sepuluh Nopember (ITS).

## REFERENCES

[1] Jahromi, N.A., Hashemi, S., Dehghantanha, A., Choo, K.K.R., Karimipour, H., Newton, D.E., Parizi, R.M.

(2020). An improved two-hidden-layer extreme learning machine for malware hunting. *Computers & Security*, 89: 101655. <https://doi.org/https://doi.org/10.1016/j.cose.2019.101655>

[2] Jha, V., Saxena, A. (2024). From code to conundrum: machine learning's role in modern malware detection. In 2024 International Conference on Advancements in Smart, Secure and Intelligent Computing, Bhubaneswar, India, pp. 1-6. <https://doi.org/10.1109/ASSIC60049.2024.10507988>

[3] Gong, D., Liu, Y. (2022). A machine learning approach for botnet detection using LightGBM. In 2022 3rd International Conference on Computer Vision, Image and Deep Learning & International Conference on Computer Engineering and Applications (CVIDL ICCEA), Changchun, China, pp. 829-833. <https://doi.org/10.1109/CVIDLICCEA56201.2022.9824033>

[4] Tuan, T.A., Long, H.V., Taniar, D. (2022). On detecting and classifying DGA botnets and their families. *Computers & Security*, 113: 102549. <https://doi.org/https://doi.org/10.1016/j.cose.2021.102549>

[5] Sharma, P., Kumar, S., Sharma, N. (2017). BotMAD: Botnet malicious activity detector based on DNS traffic analysis. In 2016 2nd International Conference on Next Generation Computing Technologies NGCT 2016, Dehradun, India, pp. 824-830. <https://doi.org/10.1109/NGCT.2016.7877524>

[6] Ibrahim, W.N.H., Anuar, S., Selamat, A., Krejcar, O., Crespo, R.G., Herrera-Viedma, E., Fujita, H. (2021). Multilayer framework for botnet detection using machine learning algorithms. *IEEE Access*, 9: 48753-48768. <https://doi.org/10.1109/ACCESS.2021.3060778>

[7] Ramesh, R.B., Thangaraj, S.J.J. (2023). Analyzing and detecting Botnet Attacks using Anomaly Detection with machine learning. In 2023 5th International Conference on Inventive Research in Computing Applications, Coimbatore, India, pp. 911-915. <https://doi.org/10.1109/ICIRCA57980.2023.10220903>

[8] Griffiths, C. (2024). The latest 2024 phishing statistics (updated June 2024). <https://aag-it.com/the-latest-phishing-statistics/>.

[9] Garg, P., Girdhar, N. (2021). A systematic review on spam filtering techniques based on natural language processing framework. In 2021 11th International Conference on Cloud Computing, Data Science & Engineering, Noida, India, pp. 30-35. <https://doi.org/10.1109/Confluence51648.2021.9377042>

[10] Eslahi, M., Salleh, R., Anuar, N.B. (2012). Bots and botnets: An overview of characteristics, detection and challenges. In IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, pp. 349-354. <https://doi.org/10.1109/ICCSCE.2012.6487169>

[11] Alshamkhany, M., Alshamkhany, W., Mansour, M., Khan, M., Dhou, S., Aloul, F. (2020). Botnet attack detection using machine learning. 2020 14th International Conference on Innovations in Information Technology (IIT), Al Ain, United Arab Emirates, pp. 203-208. <https://doi.org/10.1109/IIT50501.2020.9299061>

[12] Judyflavia, M.S.B., Sowmiyaa, P., Srianvika, S.,

- Poojitha, P. (2022). IoT botnet detection using machine learning. *International Journal of Health Sciences*. (Qassim), 6(S2): 5952-5962. <https://doi.org/10.53730/ijhs.v6nS2.6551>
- [13] Wei, C., Xie, G., Diao, Z. (2023). A lightweight deep learning framework for botnet detecting at the IoT edge. *Computers & Security*, 129: 103195. <https://doi.org/10.1016/j.cose.2023.103195>
- [14] Wang, C.Y., Ou, C.L., Zhang, Y.E., Cho, F.M., Chen, P.H., Chang, J.B., Shieh, C.K. (2018). BotCluster: A session-based P2P botnet clustering system on NetFlow. *Computer Networks*, 145: 175-189. <https://doi.org/10.1016/j.comnet.2018.08.014>
- [15] Chowdhury, S., Khanzadeh, M., Akula, R., Zhang, F., Zhang, S., Medal, H., Marufuzzaman, M., Bian, L. (2017). Botnet detection using graph-based feature clustering. *Journal of Big Data*, 4: 14. <https://doi.org/10.1186/s40537-017-0074-7>
- [16] Suryotrisongko, H., Musashi, Y. (2022). Evaluating hybrid quantum-classical deep learning for cybersecurity botnet DGA detection. *Procedia Computer Science*, 197: 223-229. <https://doi.org/https://doi.org/10.1016/j.procs.2021.12.135>
- [17] Hossain, M.A., Islam, M.S. (2023). A novel hybrid feature selection and ensemble-based machine learning approach for botnet detection. *Scientific Reports*, 13: 21207. <https://doi.org/10.1038/s41598-023-48230-1>
- [18] Rab, R., Bindu, F., Sanjida, S., Tabassoum, N., Wahab, T.B., Ghosh, A. (2023). Efficient feature selection on adversarial botnet detection. *Annals of Computer Science and Information Systems*, 37: 229-234. <https://doi.org/10.15439/2023F1425>
- [19] Ali, H.H., Yas, R.M., Abdulameer, M.H. (2023). Behavior analysis of machine learning algorithms for botnets detection. In *2023 International Conference on Information Technology, Applied Mathematics and Statistics (ICITAMS)*, Al-Qadisiya, Iraq, pp. 235-241. <https://doi.org/10.1109/ICITAMS57610.2023.10525642>
- [20] Chen, R.C., Dewi, C., Huang, S.W., Caraka, R.E. (2020). Selecting critical features for data classification based on machine learning methods. *Journal of Big Data*, 7: 52. <https://doi.org/10.1186/s40537-020-00327-4>
- [21] Ghojogh, B., Samad, M.N., Mashhadi, S.A., Kapoor, T., Ali, W., Karray, F., Crowley, M. (2019). Feature selection and feature extraction in pattern analysis: A literature review. *arXiv:1905.02845*. <https://doi.org/10.48550/arXiv.1905.02845>
- [22] Al Tawil, A., Sabri, K.E. (2021). A feature selection algorithm for intrusion detection system based on Moth Flame Optimization. In *2021 International Conference on Information Technology (ICIT)*, Amman, Jordan, pp. 377-381. <https://doi.org/10.1109/ICIT52682.2021.9491690>
- [23] Safitri, W.A., Ahmad, T., Hostiadi, D.P. (2022). Analyzing machine learning-based feature selection for botnet detection. *2022 1st International Conference on Information System & Information Technology (ICISIT)*, Yogyakarta, Indonesia, pp. 386-391. <https://doi.org/10.1109/ICISIT54091.2022.9872812>
- [24] Kalakoti, R., Nomm, S., Bahsi, H. (2022). In-depth feature selection for the statistical machine learning-based botnet detection in IoT networks. *IEEE Access*, 10: 94518-94535. <https://doi.org/10.1109/ACCESS.2022.3204001>
- [25] Lefoane, M., Ghafir, I., Kabir, S., Awan, I.U. (2023). Unsupervised learning for feature selection: A proposed solution for botnet detection in 5G networks. *IEEE Transactions on Industrial Informatics*, 19(1): 921-929. <https://doi.org/10.1109/TII.2022.3192044>
- [26] Delplace, A., Hermoso, S., Anandita, K. (2020). Cyber attack detection thanks to machine learning algorithms. *arXiv:2001.06309*. <https://doi.org/10.48550/arXiv.2001.06309>
- [27] Putra, M.A.R., Ahmad, T., Ijtihadie, R.M., Hostiadi, D.P. (2023). Detecting botnet spam activity by analyzing network traffic using two-stack decision tree algorithms. In *2023 International Conference of Computer Science and Information Technology (ICOSNIKOM)*, Binjia, Indonesia, pp. 1-6. <https://doi.org/10.1109/ICoSNIKOM60230.2023.10364480>
- [28] Putra, M.A.R., Hostiadi, D.P., Ahmad, T. (2022). Botnet dataset with simultaneous attack activity. *Data in Brief*, 45: 108628. <https://doi.org/10.1016/J.DIB.2022.108628>
- [29] Gupta, S., Singh, B. (2024). An intelligent multi-layer framework with SHAP integration for botnet detection and classification. *Computers & Security*, 140: 103783. <https://doi.org/https://doi.org/10.1016/j.cose.2024.103783>
- [30] Belkacem, S. (2024). Simultaneous botnet attack detection using long short term memory-based autoencoder and XGBoost classifier. *International Journal of Safety and Security Engineering*, 14(1): 155-163. <https://doi.org/10.18280/ijss.140115>
- [31] Belkacem, S. (2024). A comparative analysis on ensemble learning and deep learning based intrusion detection systems over the NCC2 Dataset. In *21st International Conference on Information Technology-New Generations*, Springer, Cham, pp. 111-115. [https://doi.org/10.1007/978-3-031-56599-1\\_16](https://doi.org/10.1007/978-3-031-56599-1_16)