Received: January 15, 2021. Revised: February 9, 2021. 536

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

ANOVA-SVM for Selecting Subset Features in Encrypted Internet Traffic Classification

Achmad Akbar Megantara[1] Tohari Ahmad[1]*

[1]*Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Jawa Timur, 60111, Indonesia*
* Corresponding author's Email: tohari@if.its.ac.id

Abstract: Encryption technique is widely used in the internet network for protecting user privacy, maintaining the confidentiality of the data, avoiding firewall detection, and administrating the system. To prevent encryption techniques in malicious activities such as encrypting data that contains malware or viruses, illegal transactions like selling drugs, illegal weapons and fake documents, a company or institution uses encrypted internet traffic classification to analyze and identify the activity. A challenging problem in encrypted internet traffic classification is the massive amount of data in the dataset and the existence of many irrelevant features. In this research, we propose a technique by integrating the ANOVA algorithm with the wrapper method from LinearSVC in the SVM method to overcome this problem. The ANOVA algorithm is used to analyze the data's variance, and LinearSVC to calculate the relationship between each data to its decision boundary. A new technique is proposed by calculating the mean value of the distances to remove features, which are relatively far from the decision boundary. This technique is taken to isolate features from unused ones to be used for the next process. The experimental result shows that this proposed method can compete with the existing research method and reduce system detection time. In this case, we take some research as the baseline, including that with one-dimensional convolution neural networks, over-sampling and under-sampling combination, inline and adaptive application, and FlowPic.

Keywords: Encrypted traffic classification, UNB-CIC VPN, Analysis of variance, Support vector machine, Machine learning.

1. Introduction

Along with its rapid growth, the traffic packets' encryption technique is widely used in the internet network to protect user privacy and maintain the data's confidentiality [1]. Encryption technique can also be used for allowing every user to avoid firewall detection and system administrator. Nevertheless, besides its benefit, criminals have also used this scheme to do illegal activities [2]. Some attackers implement cryptographic algorithms to encrypt the data containing malware or virus that can anonymously attack the system. It is also possible that it is also to commit illegal transactions such as buying and selling drugs, illegal weapons, and fake documents. So, network traffic encryption has become a new challenge in the cybersecurity and network management field to build a system that can identify and classify encrypted traffic data.

One technique that can be used for preventing criminal activity is network internet traffic classification, whose purpose is to categorize users' activities. This classification has some advantages, for example, trend analysis, traffic engineering, capacity planning, application performance, and anomaly detection [3].

With the rise of network technology, various applications have been developed, which involved a massively increasing number of users. It has become the main concern in this network classification process. Consequently, a more reliable and adaptive system for various changes is required.

Internet traffic classification is divided into three types: port-based [4, 5], payload-based [6-8], and machine learning-based methods [9, 10]. The port-based method is a traditional way to classify the

application/activity based on an application's port number, such as port 20 for FTP and port 80 for HTTP. However, this method is decreasingly used, as many applications started implementing a dynamic range of port allocation. The second method is the payload-based method, which utilizes the data payload to classify the application/activity. Lastly, the third is the machine learning-based method, which employs certain algorithms to train the collected data traffic activity on the internet; so that the system can detect the user activity and its application based on the trained data.

In this third method, some factors are considered in the learning stage, such as the duration of the application, consuming the time for sending data, and the anomaly in the network intrusion detection system (NIDS). Besides, machine learning in internet traffic classification lies in the significant amount of data with irrelevant features in the dataset.

These amount and irrelevant feature problems in the dataset can be overcome by performing data pre-processing, such as feature selection [11-13] and data reduction [14-16]. Feature selection is used to remove unused features (x-dimensions) from the dataset, while data reduction is to remove unused data (y-dimensions) from the dataset.

This research intends to combine the Analysis of Variance (ANOVA) method with Support Vector Machine (SVM) to improve internet traffic classification performance. ANOVA analyzes every feature's variance in the dataset, and SVM enhances the classifier's performance. The generated data from the ANOVA-SVM process is the distance value of each feature to its decision boundary, that the closer each feature to its boundary, the relevant feature it is. Each feature's distance value is distributed into two groups by calculating the mean value of the data distance value. With this proposed technique, it can be used to separate relevant features from unused features.

This paper is distributed in five parts. The first part is the background of the research. The second part is the research that relates to this study. In the third part, the proposed method is explained. The implementation result and data analysis of the proposed method are in the fourth. The last part is the conclusion drawn from this research and possible future works based on this research result.

2. Related work

Some research has worked on the encrypted traffic classification with a different research focus for the past few years, such as feature selection process, data reduction process, an imbalanced data problem, performance enhancement, accuracy improvement, and classification method. Differently, in this research, we are focusing on the feature selection process for determining features to identify or classify the user activity or application used based on traffic data. The previous research related to this study is described as follows.

Several types of research use encrypted internet traffic classification as the focus. Saber et al. [17] find imbalanced data as the main problem in the encrypted internet traffic classification dataset, reducing the system classifier's performance. To solve this problem, they offer a method that combines over and under sampling using Principal Component Analysis (PCA) and Support Vector Machine (SVM). In its implementation, they use UNB-CIC VPN Network Traffic as the dataset. Overall, this proposed method shows a satisfying result with correcting the imbalanced class problem, the most significant features are extracted, and the method shows a good performance. Meanwhile, the over-sampling and under-sampling model depends on the variance data of the dataset. A dataset with low variance is hard to implement with a sampling technique because the new data are generated from the dataset's existing data. Besides, the over-sampling and under-sampling technique generate new data and add them to the dataset. So, the total amount of data increases, which affects the system performance, especially for the computational time.

One of the problems in encrypted internet traffic classification is the performance of the system classification. Wang et al. [9] work on this issue by implementing one-dimensional convolutional neural networks. Unlike the traditional machine learning method, it is developed based on deep learning algorithms. This research used the UNB-CIC VPN Network Traffic as the dataset. Their proposed method shows a better performance than those of state-of-the-art. However, it is not designed to handle the imbalanced data problem yet and shows a worse performance in the Non-VPN traffic class.

Accuracy is one thing that is becoming a deep concern in most encryption traffic classification research. This is because it is the machine learning area's benchmark value to measure how good the system is to detect. Research conducted by Zou et al. [2] combines deep neural networks and recurrent networks to improve the classification results' accuracy. They use a convolutional network for extracting the packet features, and a recurrent network is used to pick out the flow features based on the inputs. Their experimental result shows that the method outperforms the existing state-of-the-art

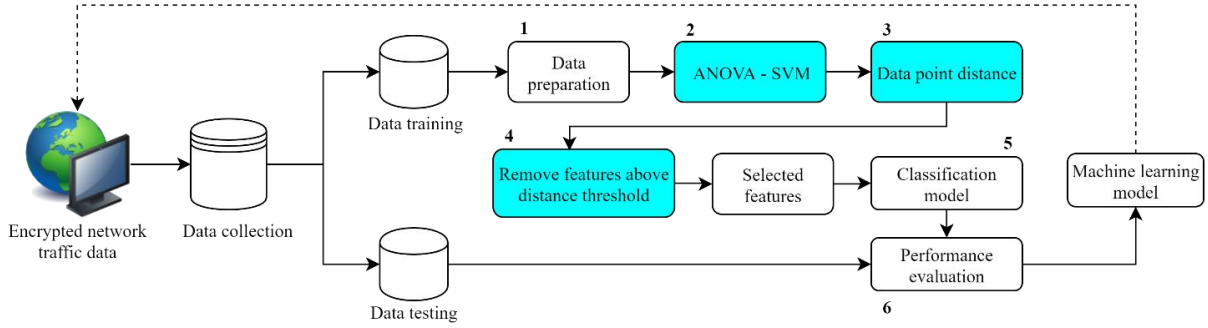

Figure. 1 The proposed method is represented in stages 2-4

model based on CNN. Because this research uses flow-based features instead of packet-based features as the model input, it requires a vast amount of data to build the model. Furthermore, each epoch's average training time is about 63.165s with 52Gb memory, which makes it difficult for the system model to be implemented in the system with low hardware specification.

Several classification methods have been used to classify the network traffic data. Research performed by Nazari et al. [18] takes the DPI-based Stream Classification Algorithm (DSCA) to build a classification method that can adaptively identify applications from network traffic data. The experimental result shows that the proposed method delivers a good performance with 96.75% and 86.92% accuracy using UNB ISCX VPN-nonVPN and UNB ISCX Tor-nonTor datasets, respectively. Nevertheless, DPI processing for the ratio traffics sent should be reduced. Moreover, this method still suffers from real high-bandwidth network data. So, the system's performance, especially for the computation time and accuracy, still needs to be improved.

3. Proposed method

This research proposes a technique by integrating the Analysis of Variance (ANOVA) algorithm with the Support Vector Machine (SVM) method. This research is inspired by that of Saber et al. [17], which uses the Principal Component Analysis (PCA) method to extract features from the dataset and Support Vector Machine (SVM) to select the features from the dataset. In this research, we use the ANOVA algorithm and the SVM method to perform the feature selection process.

As shown in Fig. 1, the dataset is put in the data preparation process, consisting of two mechanisms: data cleaning and data normalization. In the data cleaning process, the redundant data, not-a-number value, and low variance data are removed, while data normalization is used to reduce each data's scale difference. The StandardScaler method is applied for transforming the data into a range between 0 and 1.

The method proposed in this research covers from the second to the fourth stages in Fig. 1. The ANOVA-SVM method generates several parameters such as the ANOVA-SVM score, the accuracy score from each subset test, and the distance value from each data point to the decision boundary. This data point distance is the value that represents how close each feature into its decision boundary. The closer the data point to the decision boundary, the more relevant the data to its label. A threshold value is for separating features to use in the next process from unused ones, whose value is generated by calculating the mean score of the distance value from each data point. Only features below the threshold are employed in the next process; otherwise, they are removed.

The fifth and sixth stages are for evaluating the classification and finding the performance. In the classification process, a decision tree and random forest method are developed to classify and build the training model. For the performance evaluation, accuracy and time computation is calculated as shown in Fig. 1.

3.1 Analysis of Variance (ANOVA)

Analysis of Variance (ANOVA) is a statistical method introduced by Ronald Fisher to analyze the differences within the mean of data. The ANOVA test aims to determine the significance level between independent variables and the independent variables in a regression study. The ANOVA method compares more than two groups simultaneously to determine a relationship value between them [19]. The result of the ANOVA implementation, called f-statistic or f-ratio, can be used to analyze the variability between samples and within samples. The ANOVA can be calculated using Eq. (1), where F represents the

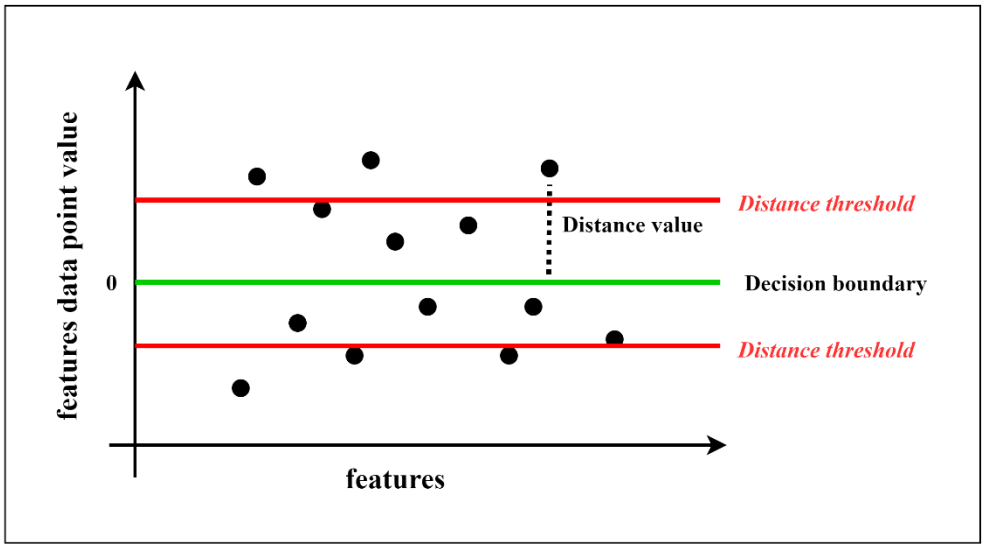

Figure. 2 Illustration of the distance value of each features data point to its decision boundary

ANOVA coefficient, MST is the mean sum of squares value, and MSE depicts the mean sum of the squares error value.

$$F = \frac{MST}{MSE} \quad (1)$$

3.2 Support vector machine (SVM)

Support Vector Machine (SVM) is a supervised learning-based method for both classification and regression problems. SVM method transforms each data point into n-dimensional features spaces, where n is the number of features in the dataset. Then, the classification process is applied to find the hyperplane between two classes [20]. The SVM method can also be used to select features by performing LinearSVC, which combines it with the ANOVA algorithm for wrapper method-based feature selection. The result of this process is the distance between each feature and its decision value/decision boundary. The illustration of the ANOVA-SVM process can be seen in Fig. 2.

$$Y = w \times x + b \quad (2)$$

$$dist = \frac{Y}{C} \quad (3)$$

For calculating this distance ($dist$), Eqs. (2) and (3) are designed, where Y is the decision boundary, w is the value of hyperplane from SVM, x represents the data points of each feature, b is the hyperplane threshold from SVM, and c is the coefficient of SVM. The mean score of the distance value from each subset is to be the distance value threshold. As mentioned earlier, features whose distance value is higher than the threshold are removed, and those that lower than the threshold are taken. The closer each feature to zero value or decision boundary, the more relevant the features.

$$thres = \frac{\sum_{i=1}^{n} dist_i}{n}, \; n \neq 0 \quad (4)$$

Eq. (4) is the formula of distance value threshold. Where $thres$ represent the distance threshold, and n is the number of features in the subset dataset.

3.3 Classification

In the classification process, Decision Tree (DT) and Random Forest Classifier (RFC) are implemented to classify the selected subset features data. The reason why using DT & RFC method as the classification method is because these methods are suitable enough for this subset dataset characteristic, such as minimum numbers of features. Besides, there is no duplicate or redundant data, and the subset dataset has high variance data.

Table 1. Data distribution in Scenario-1

SCEN – 1	**Data**			
	15	**30**	**60**	**120**
Non-VPN	8965	6917	8580	5151
VPN	9973	7734	6935	5631
Total	**18758**	**14651**	**15515**	**10782**

Table 2. Data distribution in scenario-2

SCEN - 2	Data			
	15	**30**	**60**	**120**
BROWS-ING	2500	2500	2500	2500
CHAT	890	595	778	242
STREAM-ING	482	342	252	208
MAIL	249	81	849	185
VOIP	2826	1438	1829	392
P2P	1000	1000	1000	1000
FT	1018	961	1372	624
VPN-VOIP	2271	1299	1620	386
VPN-CHAT	1196	780	514	349
VPN-STREAM-ING	2500	298	197	145
VPN-FT	1932	1158	898	716
VPN-BROWS-ING	2500	2500	2500	2500
VPN-P2P	928	851	823	813
VPN-MAIL	491	848	383	722
Total	**18758**	**14651**	**15515**	**10782**

Table 3. Data distribution in scenario-3

SCEN - 3	Data			
	15	**30**	**60**	**120**
BROWS-ING	5000	5000	5000	5000
CHAT	2086	1375	1292	591
STREAM-ING	957	640	449	353
MAIL	740	929	1232	907
VOIP	5097	2737	3449	778
P2P	1928	1851	1823	1813
FT	2950	2119	2270	1340
Total	**18758**	**14651**	**15515**	**10782**

4. Type-style and fonts

4.1 UNB-CIC VPN network traffic dataset

UNB-CIC VPN Network Traffic Dataset is developed by the University of New Brunswick in 2016 to generate a representative dataset of real-world traffic [21]. The dataset is created by capturing the users' data using specific applications such as Skype and Facebook. This dataset contains eight types of features and 14 traffic categories. The detailed information of the UNB-CIC VPN Network Traffic Dataset is in Table 1.

In this research, the dataset is divided into three scenarios. The first scenario is to distinguish non-VPN from VPN data, so the dataset is distributed in 2 classes (see Table 1). In this dataset, the first scenario is generated from the files in the scenarioA1. The second scenario is to differentiate the data between applications, which use VPN with applications and those that do not. In this scenario, the dataset is distributed in 14 classes (see Table 2). Here, the data are from the files in scenarioB. The last scenario is used to distinguish the data between each application. The data are separated into seven classes (see Table 3) by using files AllinOne in scenarioB.

4.2 Method evaluation

There are two parameters calculated in this research to check whether the proposed method can solve the dataset's existing problems. Those are accuracy and computation time. Accuracy analyzes how the proposed method can detect the data by comparing the actual data with predicted data. Computational time is used to test how fast the proposed method can classify the data in the dataset and analyze how it can reduce the computation time.

4.2.1. Performance matrix

Table 4 is the multi-class confusion matrix of this research. In the class k ($0 \leq k \leq n$), four different classification values can be generated: True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP represents the number of applications traffic data are predicted correctly in the positive class. TN represents the number of application traffic data that are predicted correctly in the negative class. FP is where the model incorrectly predicted the positive class, and FN is a value where the model incorrectly predicted the negative class.

4.2.2. Accuracy

The accuracy value refers to the predicted/measured value's closeness compared with the actual/real value. The higher the value, the better the method. Accuracy is calculated using Eq. (5).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

Table 4. Confusion matrix

		Predicted Data		
		$C_0 \ldots C_{k-1}$	C_k	$C_{k+1} \ldots C_N$
Actual Data	$C_0 \ldots C_{k-1}$	TN	FP	TN
	C_k	FN	TP	FN
	$C_{k+1} \ldots C_N$	TN	FP	TN

Table 5. Distance threshold in scenario-1

Time	Distance Value Threshold
15	±0.11336
30	±0.11063
60	±0.23565
120	±0.20777

Table 6. Selected features in scenario-1

Time	Number of Selected Features	Selected Features
15	8	'duration', 'total_biat', 'min_fiat', 'max_fiat', 'mean_fiat', 'total_fiat', 'min_biat', 'max_biat'.
30	7	'duration', 'max_biat', 'total_fiat', 'total_biat', 'min_fiat', 'min_biat', 'max_fiat'.
60	16	'std_fiat', 'std_biat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'max_flowiat', 'mean_flowiat', 'std_flowiat', 'min_active', 'mean_active', 'max_active', 'min_idle', 'max_idle', 'std_idle', 'std_active', 'mean_idle'.
120	21	'total_biat', 'min_biat', 'max_fiat', 'mean_fiat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'max_flowiat', 'mean_flowiat', 'min_active', 'mean_active', 'max_active', 'min_idle', 'mean_idle', 'std_idle', 'min_fiat', 'max_biat', 'mean_biat', 'std_flowiat', 'std_active', 'max_idle'.

4.2.3. Computational time

Computational time, also called running time, is the time needed to perform a computational process. This value can be used to represent how if the proposed method is implemented in the whole system. When it comes to the entire traffic network, the smaller the time needed to do every computation, the better the method.

However, the computational time is hardware-dependent. Different specifications of hardware used are likely to produce different results, too. In this research, the hardware specification is 8 Gb RAM, i5-3210M CPU 2.5 GHz, NVIDIA GeForce GT 630M, and Jupyter Notebook with Python 3.7.7. Computational time is calculated, starting from importing the library, building classification models, and testing it. There are two computational times compared: computational time from all features and computational time with selected features from the proposed method.

4.3 SCENARIO-1

In the first scenario, the dataset is distributed into two classes: Non-VPN and VPN. It is to identify the traffic data whether the user runs a VPN or not.

4.3.1. Selected features

Table 5 shows the distance threshold generated from the ANOVA-SVM process. The threshold is obtained by calculating the mean score from each feature's distance value in the dataset. Positive or negative values represent the class position of the data point to the decision boundary. The closer the decision value to 0, the closer it is to the decision boundary. A new subset of feature dataset selected from the experiment in scenario-1 can be seen in Table 6. The selected features are 8, 7, 16, and 21 for 15, 30, 60, and 120 datasets, respectively.

4.3.2. Accuracy score

Table 7 shows the accuracy score from Scenario-1 using the decision tree and random forest classifiers. The experimental results show that the proposed method can improve the accuracy performance by around 20 – 30%, with the highest accuracy score is 88% in dataset 15. For comparison, the other research from Wang et al. [9] gets an accuracy score above 92.3% using the 1D-CNN method with two classes VPN and Non-VPN data. Shapira and Shavitt [22], with FlowPic algorithm based on CNN method, get 85% accuracy on Non-VPN Traffic categorization, 98.4% accuracy VPN Traffic Categorization, and

Table 7. Accuracy score taken from scenario-1

Time	Accuracy (%)			
	All Features		Selected Features	
	DT	RFC	DT	RFC
15	51	58	84	88
30	58	55	82	87
60	48	54	79	84
120	50	58	81	86

Table 8. Computational time taken from scenario-1

Time	Computation Time (s)			
	All Features		Selected Features	
	DT	RFC	DT	RFC
15	0.516	7.365	0.295	5.254
30	0.300	4.080	0.125	3.295
60	0.237	3.230	0.180	3.348
120	0.140	2.830	0.130	2.030

Table 9. Distance threshold in scenario-2

Time	Distance Value Threshold
15	±0.11401
30	±0.14002
60	±0.25268
120	±0.19037

67.8% accuracy for Tor Traffic Categorization. Nazari et al. [18], with DSCA or DPI-based Stream Classification Approach, gets accuracy score above 90% on the VPN dataset and 87% accuracy on the Tor dataset. The accuracy score of the proposed research is competitive enough compared with other research.

Unlike other studies, our proposed research is in the pre-processing step of the feature selection process, while the other research focuses on the processing step. The difference between them is that if it is located in the processing step, the method will affect the machine performance, and it will depend on the hardware specification. Because our proposed research is located on the pre-processing step, it will not affect the machine processing performance and can be applied to a system with different hardware specifications.

4.3.3. Computational time

Table 8 is the computational time data obtained from the Scenario-1. It is calculated from the import library step into testing the model. The experimental results show that the proposed method can reduce the computational time to 1 – 2 s for each process. The computational time is calculated from the import library process to the classification process. The fewer features cause the computational time reduction in the dataset, and subset features obtained consist of important/relevant selected features.

Table 10. Selected features in scenario-2

Time	Number of Selected Features	Selected Features
15	17	'max_biat', 'mean_biat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'max_flowiat', 'mean_flowiat', 'std_flowiat', 'min_active', 'mean_active', 'max_active', 'mean_fiat', 'std_active', 'mean_idle', 'max_idle', 'min_flowiat', 'std_idle', 'min_idle'.
30	18	'max_fiat', 'mean_fiat', 'mean_biat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'mean_flowiat', 'std_flowiat', 'min_active', 'max_active', 'min_idle', 'mean_idle', 'std_idle', 'max_biat', 'max_flowiat', 'mean_active', 'std_active', 'max_idle'
60	19	'max_biat', 'mean_biat', 'std_fiat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'mean_flowiat', 'min_active', 'max_active', 'min_idle', 'max_idle', 'mean_fiat', 'std_biat', 'max_flowiat', 'std_flowiat', 'mean_active', 'std_active', 'mean_idle', 'std_idle'
120	21	total_biat', 'min_biat', 'max_biat', 'mean_fiat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'max_flowiat', 'mean_flowiat', 'std_flowiat', 'min_active', 'max_active', 'min_idle', 'max_idle', 'min_fiat', 'max_fiat', 'mean_biat', 'mean_active', 'std_active', 'mean_idle', 'std_idle'

4.4 SCENARIO-2

In scenario-2, the dataset is distributed into 14 classes: BROWSING, CHAT, STREAMING, MAIL, VOIP, P2P, FT, VPN-VOIP, VPN-STREAMING, VPN-FT, VPN-BROWSING, VPN-P2P, VPN-MAIL. It is to separate the applications that used a VPN from those that do not.

4.4.1. Selected features

Table 9 is the distance threshold for each dataset in Scenario-2. Similar to the experiment in scenario-2, the distance value is obtained by calculating the mean score from each dataset's distance value. The experimental results show that the number of features selected from the experiment in scenario-2 is 17, 18, 19, and 21 for 15, 30, 60, and 120 datasets, respectively. The selected features from the experiments in scenario-2 can be seen in Table 10.

4.4.2. Accuracy score

Table 11 shows the accuracy score from experimental results in the Scenario-2. Based on this table, we find the proposed method can improve the system's accuracy performance from around 30% to 50% for each dataset in scenario-2 with the highest score of accuracy is 79%. However, the proposed method still needs more improvement to compete with the other research, such as that from Saber et al. [17] that used PCA-SVM to classify 14-classes encrypted traffic data whose highest accuracy score is 96.6%.

4.4.3. Computational time

Table 12 is the computational time data from the Scenario-2. Like the experiment in Scenario-1, the computational time is obtained by calculating the length of running time from the import library step to testing the model. The experimental result shows that the proposed method can reduce the computational time around 1 – 2 s for each dataset in Scenario-2.

4.5 SCENARIO-3

In Scenario-3, the dataset is distributed into seven classes: BROWSING, CHAT, STREAMING, MAIL, VOIP, P2P, and FT. The purpose of scenario-3 is to identify the applications that the user runs. In this scenario, we can identify the running application in the network by analyzing its traffic data's behavior.

4.5.1. Selected features

Table 13 is the data of the distance value threshold from the Scenario-3. There are positive and negative values from the data; these positive and negative values represent the data point's position to the decision boundary. The experimental results show that the number of features selected is 20, 18, 16, and 21 for the 15, 30, 60, and 120 datasets in Scenario-3. Selected features from scenario-3 can be seen in Table 14.

Table 11. Accuracy score taken from scenario-2

Time	Accuracy (%)			
	All Features		Selected Features	
	DT	RFC	DT	RFC
15	30	17	73	79
30	25	34	72	78
60	27	35	69	77
120	21	32	68	75

Table 12. Computational time taken from scenario-2

Time	Computational Time (s)			
	All Features		Selected Features	
	DT	RFC	DT	RFC
15	0.625	8.260	0.409	7.400
30	0.354	5.363	0.271	5.405
60	0.412	6.094	0.327	6.555
120	0.271	3.742	0.266	4.064

Table 13. Distance value threshold of scenario-3

Time	Distance Value Threshold
15	±0.08559
30	±0.03506
60	±0.13896
120	±0.03544

4.5.2. Accuracy score

Table 15 is the accuracy score from the experiment in Scenario-3. It shows that the proposed method can improve the performance of the decision tree and random forest classifiers for detecting the applications from around 30 – 60% for each dataset in Scenario-3 that the highest accuracy score can be obtained is 86%.

4.5.3. Computational time

Table 16 is the computational time from the experiment in Scenario-3. It is found that the proposed method can reduce the computational time needed for each process.

Table 14. Selected features in scenario-3

Time	Number of Selected Features	Selected features
15	20	min_fiat', 'max_fiat', 'max_biat', 'mean_biat','flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'mean_flowiat', 'std_flowiat', 'min_active', 'max_active', 'min_idle', 'mean_idle', 'std_idle', 'min_biat', 'mean_fiat', 'max_flowiat', 'mean_active', 'std_active', 'max_idle'
30	18	'max_fiat', 'mean_fiat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'max_flowiat', 'std_flowiat', 'min_active', 'max_active', 'min_idle', 'mean_idle', 'std_idle', 'mean_active', 'max_biat', 'mean_biat', 'mean_flowiat', 'std_active', 'max_idle'
60	16	'std_fiat', 'std_biat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'mean_flowiat', 'min_active', 'max_active', 'min_idle', 'max_idle', 'max_flowiat', 'std_flowiat', 'mean_active', 'std_active', 'std_idle', 'mean_idle'
120	21	'total_biat', 'min_biat', 'max_fiat', 'mean_biat', 'flowPktsPerSecond', 'flowBytesPerSecond', 'min_flowiat', 'max_flowiat', 'mean_flowiat', 'min_active', 'max_active', 'std_active', 'min_idle', 'std_idle', 'min_fiat', 'max_biat', 'mean_fiat', 'std_flowiat', 'mean_active', 'mean_idle', 'max_idle'

Table 15. Accuracy score taken from scenario-3

Time	Accuracy (%)			
	All features		Selected features	
	DT	RFC	DT	RFC
15	24	26	81	86
30	24	26	81	86
60	35	58	80	84
120	45	55	79	85

Table 16. Computational time taken from scenario-3

Time	Computational Time (s)			
	All Features		Selected Features	
	DT	RFC	DT	RFC
15	0.34	4.100	0.275	3.860
30	0.33	3.160	0.195	2.950
60	0.238	2.733	0.169	2.631
120	0.180	2.330	0.140	2.030

5. Conclusion

Based on the results of the experiment, we get some information as follows. First, the distance is a value that represents how close each feature to the decision boundary. It can be used as the parameter to measure how relevant features to its label. To separate those relevant features from non-relevant ones, the threshold value is calculated using the mean score from each dataset's distance value.

The benefit of our proposed research is on the threshold value mechanism. With this proposed method, we can automatically set the threshold to isolate the relevant features used in the next process and irrelevant features that will be removed. The proposed research is located in the pre-processing step, especially in the feature selection. After comparing with other research, we find that this method is not affected by the machine performance and can be applied in the system with different specifications. Furthermore, this method can increase the accuracy with various values, depending on the dataset's characteristics and the respective scenario. For example, by selecting relevant features, the proposed method can achieve 88% of accuracy, which is significantly higher than that without feature selection.

Second, this research shows that the proposed method can improve the detection's accuracy performance using decision tree and random forest classifier methods. The experimental results prove that the proposed method can improve the accuracy score and get a competitive score compared to some existing research. Third, using only selected features, the computational time can be reduced. This reduction has made it more possible to implement in the real environment.

In the next research, the accuracy score may be improved further by implementing several methods such as data reduction and optimization. There is also a possibility to make it adaptive in various environments.

Conflicts of Interest

The authors declare that they have no competing interests.

Author Contributions

Conceptualization, research methodology, implementation and experiments, writing—original draft preparation: AAM.

Supervision, writing—review, editing, administration of research and funding: TA.

References

[1] M. Faheem, S. Jamel, A. Hassan, Z. A., N. Shafinaz, and M. Mat, "A Survey on the Cryptographic Encryption Algorithms", *Int. J. Adv. Comput. Sci. Appl.*, Vol. 8, No. 11, pp. 333–344, 2017.

[2] Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, and Z. Yao, "Encrypted Traffic Classification with a Convolutional Long Short-Term Memory Neural Network", In: *Proc. of - 20th Int. Conf. High Perform. Comput. Commun. 16th Int. Conf. Smart City 4th Int. Conf. Data Sci. Syst. HPCC/SmartCity/DSS 2018*, pp. 329–334, 2019.

[3] B. Yamansavascilar, M. A. Guvensan, A. G. Yavuz, and M. E. Karsligil, "Application identification via network traffic classification", In: *Proc. of 2017 Int. Conf. Comput. Netw. Commun. ICNC 2017*, pp. 843–848, 2017.

[4] P. Schneider, "TCP/IP Traffic Classification Based on Port Numbers", *Schneidergrinch*, pp. 2–7, 1992.

[5] G. Cheng and S. Wang, "Traffic classification based on port connection pattern", In: *Proc. of 2011 Int. Conf. Comput. Sci. Serv. Syst. CSSS 2011 - Proc.*, pp. 914–917, 2011.

[6] F. Dehghani, N. Movahhedinia, M. R. Khayyambashi, and S. Kianian, "Real-time traffic classification based on statistical and payload content features", In: *Proc. of - 2010 2nd Int. Work. Intell. Syst. Appl. ISA 2010*, pp. 26–29, 2010.

[7] H. K. Lim, J. B. Kim, K. Kim, Y. G. Hong, and Y. H. Han, "Payload-based traffic classification using multi-layer LSTM in software defined networks", *Appl. Sci.*, Vol. 9, No. 12, 2019.

[8] F. Risso, M. Baldi, O. Morandi, A. Baldini, and P. Monclus, "Lightweight, payload-based traffic classification: An experimental evaluation", In: *Proc. of IEEE Int. Conf. Commun.*, pp. 5869–5875, 2008.

[9] W. Wang, M. Zhu, J. Wang, X. Zeng, and Z. Yang, "End-To-end encrypted traffic classification with one-dimensional convolution neural networks", In: *Proc. of 2017 IEEE Int. Conf. Intell. Secur. Informatics Secur. Big Data, ISI 2017*, pp. 43–48, 2017.

[10] Y. Hou, H. Huang, W. Shao, and H. Huang, "Traffic classification method by combination of host behaviour and statistical approach", *J. Eng. Sci. Technol. Rev.*, Vol. 7, No. 3, pp. 151–157, 2014.

[11] P. Maniriho, L. J. Mahoro, E. Niyigaba, Z. Bizimana, and T. Ahmad, "Detecting intrusions in computer network traffic with machine learning approaches", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 3, pp. 433–445, 2020.

[12] B. N. Kumar, M. S. V. S. B. Raju, and B. V. Vardhan, "Enhancing the performance of an intrusion detection system through multi- linear dimensionality reduction and Multi-class SVM", *International Journal of Intelligent Engineering and Systems*, Vol. 11, No. 1, pp. 181–192, 2018.

[13] A. Pasyuk, E. Semenov, and D. Tyuhtyaev, "Feature Selection in the Classification of Network Traffic Flows", In: *Proc. of 2019 Int. Multi-Conf. Ind. Eng. Mod. Technol. FarEastCon 2019*, pp. 1–5, 2019.

[14] A. N. Iman and T. Ahmad, "Data Reduction for Optimizing Feature Selection in Modeling Intrusion Detection System", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 6, pp. 199–207, 2020.

[15] Q. V. Dang, "Outlier detection on network flow analysis", *arXiv*, pp. 2–4, 2018.

[16] M. R. Batchanaboyina and N. Devarakonda, "Efficient outlier detection for high dimensional data using improved monarch butterfly optimization and mutual nearest neighbors algorithm: IMBO-MNN", *International Journal of Intelligent Engineering and Systems*, Vol. 13, No. 2, pp. 63–73, 2020.

[17] A. Saber, B. Fergani, and M. Abbas, "Encrypted Traffic Classification: Combining Over-and Under-Sampling through a PCA-SVM", In: *Proc. of - PAIS 2018 Int. Conf. Pattern Anal. Intell. Syst.*, pp. 1–5, 2018.

[18] Z. Nazari, M. Noferesti, and R. Jalili, "DSCA: An inline and adaptive application identification approach in encrypted network traffic", In: *Proc. of ACM Int. Conf. Proc. Ser.*, No. January, pp. 39–43, 2019.

[19] E. Ostertagová and O. Ostertag, "Methodology and Application of Oneway ANOVA", *Am. J. Mech. Eng.*, Vol. 1, No. 7, pp. 256–261, 2013.

[20] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive

survey on support vector machine classification: Applications, challenges and trends", *Neurocomputing*, Vol. 408, pp. 189–215, 2020.

[21] G. Draper-Gil, A. H. Lashkari, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of encrypted and VPN traffic using time-related features", *ICISSP 2016* - In: *Proc. of 2nd Int. Conf. Inf. Syst. Secur. Priv.*, No. February, pp. 407–414, 2016.

[22] T. Shapira and Y. Shavitt, "FlowPic: Encrypted Internet Traffic Classification is as Easy as Image Recognition", In: *Proc. of INFOCOM 2019 - IEEE Conf. Comput. Commun. Work. INFOCOM WKSHPS 2019*, pp. 680–687, 2019.