International Journal of Electrical and Computer Engineering (IJECE)
Vol. 11, No. 3, June 2021, pp. 2360~2370
ISSN: 2088-8708, DOI: 10.11591/ijece.v11i3.pp2360-2370

Optimization of network traffic anomaly detection using machine learning

Cho Do Xuan[1], Hoang Thanh[2], Nguyen Tung Lam[3]
[1]Information Security Department, Posts and Telecommunications Institute of Technology, Hanoi, Vietnam
[2]Digital Marketing, FPT University, Hanoi, Vietnam
[3]Information Assurance Department, FPT University, Hanoi, Vietnam

Article Info

Article history:

Received Sep 6, 2020
Revised Dec 9, 2020
Accepted Dec 15, 2020

Keywords:

Feature optimization
Machine learning
Network traffic
Network traffic anomaly detection
Optimization

ABSTRACT

In this paper, to optimize the process of detecting cyber-attacks, we choose to propose 2 main optimization solutions: Optimizing the detection method and optimizing features. Both of these two optimization solutions are to ensure the aim is to increase accuracy and reduce the time for analysis and detection. Accordingly, for the detection method, we recommend using the Random Forest supervised classification algorithm. The experimental results in section 4.1 have proven that our proposal that use the Random Forest algorithm for abnormal behavior detection is completely correct because the results of this algorithm are much better than some other detection algorithms on all measures. For the feature optimization solution, we propose to use some data dimensional reduction techniques such as information gain, principal component analysis, and correlation coefficient method. The results of the research proposed in our paper have proven that to optimize the cyber-attack detection process, it is not necessary to use advanced algorithms with complex and cumbersome computational requirements, it must depend on the monitoring data for selecting the reasonable feature extraction and optimization algorithm as well as the appropriate attack classification and detection algorithms.

Corresponding Author:

Cho Do Xuan
Department of Information Security
Posts and Telecommunications Institute of Technology
122 Hoang Quoc Viet, Cau Giay District, Hanoi, Vietnam
Email: chodx@ptit.edu.vn

1. INTRODUCTION

The cyber-attack is a form of dangerous attack that has increased rapidly in both the number of recorded attacks and the extent of their damage to organizations and businesses. The research [1-3] classified cyber-attack techniques into two main methods: Passive attack and active attack. According to the report [4], in 2019, cyber-attack techniques are considered as the top of the most dangerous attack techniques. From the statistics about security vulnerabilities [5] that are often exploited in the system by attackers, we can see the level and the danger of current cyber-attacks for organizations, governments, and businesses. Therefore, the problem of detecting and early warning signs of cyber-attack campaigns is very necessary today.

The studies [2, 3] presented the difference between cyber-attack and other attack techniques, thus making the detection and the warning of this attack have many difficulties. Currently, there are two main methods for detecting cyber-attacks: signature-based method through the rule sets, and anomaly-based method based on data analysis and statistics to find out abnormal characteristics in the network [1-3, 6]. The

signature-based method has the ability to detect quickly and accurately but cannot detect new attack techniques [1]. The anomaly-based method is not only capable of detecting attacks but also capable of detecting abnormal behaviors, but this method requires complex calculation and processing, and has low accuracy. The anomaly-based method is usually based on two main techniques that are machine learning and deep learning to classify abnormal and normal behavior [1, 2]. In this paper, we propose a cyber-attack detection method using the random forest (RF) machine learning algorithm. The RF algorithm has been proved as the current best algorithm for classification by studies [1, 3, 6-8]. The study [1, 2] listed and analyzed some data sets commonly used for cyber-attack detection such as DARPA/KDD Cup99, CAIDA, NSL-KDD, ISCX 2012, UNSW-NB15, etc. In these datasets, the UNSW-NB15 data set is built and developed relatively in accordance with real network systems [1, 9]. Therefore, in this paper, we will use the UNSW-NB15 dataset to experiment with cyber-attack detection methods.

As presented above, in order to optimize the process of detecting and alerting cyber-attacks based on machine learning techniques, recent studies and recommendations often attempt to find new detection methods and techniques. However, we recognize that the new approaches are usually only suitable for existing datasets, when they are applied in practice, they often don't bring high efficiency due to the incompatibility of model building datasets with monitoring datasets. Therefore, in our point of view, instead of trying to learn or develop new detection methods, we look for ways to analyze and build experimental datasets so that they are most suitable for real network monitoring systems. In this paper, in order to optimize the abnormal detection process based on the UNSW-NB15 dataset, we propose methods of evaluating and selecting new features. The methods that we propose to use in this paper include information gain, principal component analysis, and correlation coefficient method.

Our research is presented as follows: the urgency of the research problem is presented in section 1. In section 2, we present the process of researching, surveying, and evaluating related works. The algorithms related to the problem of classifying attack and reducing feature dimensions are presented in section 3. Section 4 presents the results of the experimental process. Accordingly, section 4.1 is the experimental process of detecting cyber-attacks, in which we evaluate and compare our proposed method with some other studies. The results of the process of evaluating and comparing the efficiency of the feature dimension reduction method are presented in section 4.2. Conclusion and evaluation are presented in section 5. The practical significance and scientificity of our paper include:

- Apply RF machine learning algorithm and UNSW-NB15 dataset to detect abnormal behavior in the network. In the studies that we surveyed (see Section 2.1), the authors used different machine learning methods to compare and evaluate the effectiveness of each algorithm. However, no research has applied the RF algorithm to detect anomalies based on the UNSW-NB15 data set, although this algorithm has been indicated as the current best algorithm for classification by some studies. Our experimental results presented in section 4.1 prove the effectiveness of RF algorithm in detecting anomalies and show that when building abnormal detection systems, it is not necessary to set up algorithms that are too cumbersome and complicated. In addition, based on the results of our proposed experimental scenarios, we have shown the options for selecting the dataset and parameters of the algorithm so that they are in compliance with the detection model.
- Proposing methods of evaluating and selecting features. In this paper, we propose to use some methods and techniques in order to evaluate and select the best features. In addition, we will reassess the detection model based on the selected features with two criteria: accuracy and processing time. The results of the research and evaluation in section 4.2 are developments and supplements to the shortcomings of the studies presented in section 2.2.

2. RELATED WORK

2.1. Cyber-attacks detection based on UNSW-NB15 dataset

In the study [10], Kumar *et al.* proposed a method to classify cyber-attack techniques based on UNSW-NB15 by using different rule sets. However, in this study, building and applying the rule set will be limited because the coverage and the number of rule sets are not large enough. Moustafa *et al.* [11] proposed the geometric area analysis technique to detect cyber-attacks by using trapezoidal area estimation. To evaluate the effectiveness of the proposed method, the authors conducted experiments on UNSW-NB15 and NSL-KDD datasets. Experimental results in the study showed the superiority of the UNSW-NB15 dataset over the NSL-KDD dataset. Besides, research [12] presents a technique for building an effective anomaly detection system based on two datasets: the NSL-KDD and UNSW-NB15. This technique requires three modules: capturing and logging module, pre-processing module, and the Dirichlet mixture model that is a novel statistical decision engine based on anomaly detection technique. The first module scans and gathers network data. Then the second module analyzes and filters these data in order to improve the efficiency of

the decision engine. Finally, the decision engine is built based on the Dirichlet mixture model. Bagui *et al.* [13] proposed the cyber-attacks detection method based on Naïve Bayes, and decision trees (J48) algorithm. In their experimental section, the research team [13] used these algorithms in turn to classify different cyber-attack components in the UNSW-NB15 dataset. In the study [14], the authors proposed a model to detect cyber-attacks using stacking techniques. Accordingly, in the training process of their model, the author uses machine learning algorithms consisting of K-Nearest Neighbors, Decision Tree, and Logistic Regression in order to build a model based on the UNSW-NB15 and UGR'16 datasets. The study [15] evaluated the effectiveness of 8 machine learning algorithms (consisting of 2-layer and 3-layer algorithms) for network intrusion detection. This is a good idea, but it requires the use of the Microsoft Azure Machine Learning Studio system to apply in practice. In this research, we proceeded to distinguish between attack and normal based on pure machine learning algorithms and the use of Apache Spark technology. Our results are similar to the results of the method that authors [15] proposed, but our performance and experimental configuration are much simpler than the research [15].

In addition, other studies also presented methods to detect attack components in the network using machine learning algorithms. The study [16] presented a method of detecting DDOS attacks using a technique that comprehensively simulates DDOS attacks. In their study [17], Narender *et al.* proposed a method to detect DDOS attacks using machine learning algorithms such as Logistic Regression, Decision Tree, and K-Nearest Neighbors. This is a relatively classic approach. Nowadays, these classification algorithms are often not as effective as the RF algorithm [7]. Jafar *et al.* [18] proposed a method to classify DOS, Prob, U2R, and L2R attack techniques by using some algorithms consisting of Neural Network, Genetic, and Decision Tree. However, the approach using classification algorithms with KDD 99 dataset in the study is an old one because the current cyber-attack data is much more abundant and diverse.

2.2. The problem of optimizing the anomaly detection feature on the network based on the UNSW-NB15 dataset

In the study [19], the author proposed using Pearson's correlation coefficient and gain ratio technique to evaluate features. However, the limitation of this study is that the authors didn't conduct experiments to evaluate the accuracy of each method of feature dimension reduction. In this paper, we will not only evaluate features to select important features but also evaluate the anomaly detection model based on the feature evaluation process. The study [20] proposed the Information gain method to reduce the feature dimension in the training process of the botnet detection model. However, in that study, the authors didn't specify which redundant features were removed. The study [10] described the Information gain algorithm for reducing the feature dimension. However, in the experimental part, the authors didn't compare the effectiveness of the detection method when using the feature dimension reduction technique. Bagui *et al.* [13] proposed methods of feature selection using K-means Clustering and Correlation based Feature Selection algorithms. In the study [21], the authors proposed using a deep learning model combining Convolutional Neural Network and long short-term memory network (LSTM) to extract and classify cyber-attacks using the CICIDS2017 dataset. Experimental results show that the classification system gives overall accuracy as 98.67% and the accuracy of each attack type as over 99.50%. However, this approach requires a lot of time and a cumbersome calculation system. Thus this method is only suitable for studies and is difficult to apply in reality.

3. ANOMALY CLASSIFICATION AND ITS OPTIMIZATION USING MACHINE LEARNING

3.1. Experimental data

The data set used for experiments is UNSW-NB15. This dataset was built by using the IXIA PerfectStorm tool to extract a mixture of attack operations in the network. More than 100 GB of raw network traffic are captured by Tcpdump tool and processed by Argus, Bro-IDS, and twelve algorithms written in the C# language to extract 43 features and save it in CSV format [9, 10, 12, 13]. The selected features are divided into six groups:

- Flow features: Include features used to identify network flow such as IP address, port number, and protocol.
- Basic features: Include connection description features.
- Content features: Consist of features of TCP/IP protocol, and features of HTTP application layer protocol.
- Time features: include time-related features such as packet arrival time, start/end time and round trip time of TCP protocol.
- Additional generated features. Features in this group can be divided into two smaller groups: general purpose features and connection features.
- Labeled features: are labels for records.

3.2. Anomaly classification using random forest machine learning algorithm

The study [7] surveyed and evaluated some supervised learning algorithms in the cyber-attack detection problem. Accordingly, the study indicates that the RF algorithm is the current best classification technique. Therefore, in this paper, we will use the RF algorithm to detect anomalies in the network based on the UNSW-NB15 dataset. RF is an ensemble classification method [22]. This algorithm is based on an ensemble of classifiers, which normally are decision trees to make the final prediction [23]. The theoretical foundation of this algorithm is based on Jensen's inequality [23]. According to Jensen's inequality applied to the classification problems, it is shown that the combination of many models may produce less error rate than that of each individual model.

3.3. Feature evaluation and selection

In fact, not all features, which we found, are useful to build a training model to help make the necessary predictions. Using a few features sometimes reduces the accuracy of prediction and takes time to build a model. Therefore, feature selection plays a very important, necessary role in the process of building abnormal detection systems. Selecting good features will not only improve the accuracy of attack prediction but also reduce feature extraction time. In this paper, we evaluate and select features by some different methods in order to assess the effectiveness of each method for the UNSW-NB15 dataset.

3.3.1. Feature optimization using correlation coefficient method

The correlation coefficient is a statistical index that measures the strength of the relationship between two variables. There are many different kinds of correlation coefficients. In this paper, we used the Pearson correlation coefficient. Pearson correlation coefficient between two variables *X* and *Y* is calculated by the formula [24].

$$\rho_{X,Y} = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$$

where:
Cov (X, Y) is the covariance of *X* and *Y*
σ_X is the standard deviation of *X*
σ_Y is the standard deviation of *Y*

The correlation coefficient has a value between -1 and 1. The negative correlation coefficient indicates that the two variables have a negative correlation or inverse correlation (is a perfect negative correlation when the value is -1). The positive correlation coefficient indicates a positive correlation (is a perfect positive correlation when the value is 1). The correlation coefficient is zero if two variables are independent of each other. Features with large correlation coefficients have linear dependence, and thus they have almost the same effect on the dependent features. So we can reduce one of those two features.

3.3.2. Feature optimization using information gain method

Information gain (IG) is a feature evaluation method based on entropy function and is widely used in machine learning [25]. Information gain is defined as a quantity that measures the amount of information gained about a class from a feature. Information gain is calculated based on entropy quantity [23]. The entropy function is defined as follows [23]: Given a probability distribution of a discrete variable x can receive n different values $\{x_1, x_2, \ldots, x_n\}$. Suppose that the probability for x get these values are $p_x = p(x = x_i)$ with $0 \le p_i \le 1$ and $\sum_{i=1}^{n} p_i = 1$. This distribution symbol is $P = (p_1, p_2, \ldots, p_n)$. The entropy of this distribution is defined by formula (1)

$$H(P) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad (1)$$

From the formula of entropy, we formulate the calculation principle of Information gain as follows:

Step 1: Consider a problem with C different classes. Suppose that we work with a non-leaf node with data points forming the set S with the number of elements as $S \vee N$. Suppose further that in these N data points, there are N_c points (with $c = 1,2,\ldots,C$) belongs to class c. The probability for each data point belongs to class c is approximately $\frac{N_c}{N}$ (maximum likelihood estimation). Thus, the entropy at this node is calculated as follows:

$$H(S) = -\sum_{c=1}^{C} \frac{N_c}{N} \log_2 \frac{N_c}{N} \quad (2)$$

Step 2: Assuming that the dataset is divided into subsets according to a feature x. Based on x, data points in S are divided into Kchild nodes: S_1, S_2, …, S_K with m_1, m_2, …, m_K points in each child node. We define formula (3) as the sum of weighted entropy of each child node. The taking weight is important because nodes often have different the numbers of points.

$$H(x,S) = -\sum_{k=1}^{K} \frac{m_k}{N} H(S_k) \quad (3)$$

Step 3: Calculate information gain value based on feature x.

$$G(x,S) = H(S) - H(x,S) \quad (4)$$

3.3.3. Feature optimization using principal component analysis method

Principal component analysis (PCA) is a method of finding a new basis so that the information of the data is mainly concentrated in several coordinates, the remainder only contains a small amount of information. To simplify the calculation, PCA will look for an orthonormal basis to make a new basis so that in this system, the most important components are in some coordinates of the first component [26]. We can see the steps for implementing PCA as follows [26, 27]:

Step 1: Calculate the mean vector of all data.

$$\bar{X} = \frac{1}{N}\sum_{n=1}^{N} X_n \quad (5)$$

Step 2: Subtract the mean vector from each data point.

$$\hat{X} = X_n - \bar{X}_n \quad (6)$$

Step 3: Calculate the covariance matrix:

$$S = \frac{1}{N}\hat{X}\hat{X}^T \quad (7)$$

Step 4: Calculate eigenvalues and eigenvectors with norm equal to 1 of this matrix, arrange them in the descending order of eigenvalues.

Step 5: Select K eigenvectors with K highest eigenvalues to build the matrix U_K whose columns form an orthogonal. These K vectors are also called key components that form a subspace close to the distribution of the normalized original data.

Step 6: Project the normalized original data $\widehat{X}$ down to the found subspace.

Step 7: Calculate the coordinates of the new data. The new data is the coordinates of the data points on the new space according to the formula (8).

$$Z = U_K^T \widehat{X} \quad (8)$$

The original data can be approximated according to the new data as in formula (9).

$$X \approx U_K Z + \bar{X} \quad (9)$$

4. EXPERIMIMENTS AND EVALUATIONS

4.1. Experiment and evaluation of abnormal detection method

4.1.1. Experimental scenarios

The experimental dataset in our paper includes 2,540,047 records consisting of 2,218,764 normal records and 321,283 attack records. We will divide the above dataset into experimental datasets as follows:

- Dataset A: consist of 322,106 normal records and 321,283 abnormal records.
- Dataset B: consist of 964,971 normal records and 321,283 abnormal records.
- Dataset C: consist of 2,218,764 normal records and 321,283 abnormal records.

Each small dataset above is divided into two parts in a ratio of 7:3 to conduct training and testing. For the classification algorithm, to evaluate the effectiveness of the RF algorithm on each dataset A, B, C, we change the parameters representing the number of decision trees in the RF algorithm. The model will be

tested with the number of decision trees used as {10, 40, 60, 80, 100}. Besides, we also conduct experiments to compare the RF algorithm with some algorithms of other studies including decision tree (J48) [9, 21] and LSTM [21, 28] algorithms. In the study [15], the authors have proven that the KNN and logistic regression algorithms both have less efficiency than the decision tree algorithm, so to see the effectiveness of the RF algorithm, we will only compare it with decision tree and LSTM algorithms

4.1.2. Evaluation criteria

In this paper, we specify that the abnormal record is labeled as *positive***,** and normal records are labeled as *negative.* The metrics used to evaluate the effectiveness of the abnormal detection method in our paper include:

- Accuracy: the ratio between the number of points correctly predicted and the total number of points in the test dataset.

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \quad (10)$$

- Precision: the ratio of the number of true positive points among those classified as *positive* (TP+FP). High Precision value means that the accuracy of the found points is high.

$$precision = \frac{TP}{TP + FP} \times 100\% \quad (11)$$

- Recall is defined as the ratio of the number of true positive points among those that are actually *positive* (TP+FN). High recall value means that the true positive rate (TPR) is high meaning that the rate of missing the actual positive points is low.

$$\text{Re}\, call = \frac{TP}{TP + FN} \times 100\% \quad (12)$$

In which, True positive (TP) is the number of abnormal records that are correctly predicted; False positive (FP) is the number of normal records that are incorrectly predicted; True negative (TN) is the number of normal records correctly predicted; False negative (FN) is the number of abnormal records that are incorrectly predicted.

- Confusion matrix: This matrix will show how many data points actually belong to which class and how many data points are predicted to belong to which class. In addition, the TPR, FNR, FPR, TNR (R-Rate) criteria are calculated based on the normalized confusion matrix. Table 1 describes the calculation formulas of the above parameters.

Table 1. Confusion matrix

	Predicted as abnormal	Predicted as normal
Actual abnormal	$TPR = TP/(TP + FN)$	$FNR = FN/(TP + FN)$
Actual normal	$FPR = FP/(FP + TN)$	$TNR = TN/(FP + TN)$

4.1.3. Experimental results

a. Experimental results with dataset A

From Table 2, we can see that when the number of decision trees is 40, the algorithm has the highest accuracy and precision which are 99.299% and 98.619% respectively. Besides, when changing the number of decision trees from 10 to 100, the accuracy of the algorithm doesn't change much. This shows that with a dataset balanced about the ratio of normal and abnormal records, the RF algorithm detects well and steadily. However, when the number of decision trees increases, training and testing time also increases. Table 3 shows the evaluation result of the confusion matrix in case of the number of decision trees of 40. From Table 3, we can see that the prediction model achieved very high accuracy in both normal and anomaly predictions.

b. Experimental results with dataset B

From Table 2 and Table 4, we can see that the accuracy and precision of dataset B are lower than the dataset A. However, the recall values don't change much. In addition, the training time of dataset B is 1.3 to 1.5 times higher than the dataset A. For the RF algorithm in dataset B, the highest accuracy (98.944%) is achieved when the number of decision trees is 60 and 80. However, the highest precision (95.965%) is

achieved in case of the number of decision trees of 40. Table 5 shows the result of the confusion matrix in case of the number of decision trees of 40.

Table 2. Experimental results with dataset a using RF algorithm

Algorithm		Accuracy %	Precision%	Recall %	FPR %	TNR %	FNR %	Training time (s)
Random Forest with the number of trees as	10	99.298	98.617	99.996	1.399	98.601	0.004	111.051
	40	99.299	98.619	99.997	1.397	98.603	0.003	117.402
	60	99.296	98.614	99.996	1.402	98.598	0.004	125.954
	80	99.299	98.618	99.997	1.398	98.602	0.003	131.755
	100	99.296	98.615	99.994	1.401	98.599	0.006	138.924
J48 [12, 20]		98.473	89.242	99.981	1.745	98.255	0.019	98.209
LSTM [20, 27]		97.682	96.888	98.522	3.156	96.844	1.478	165.453

Table 3. Confusion matrix result with the number of decision trees of 40 on dataset A

	Predicted as abnormal	Predicted as normal
Actual abnormal	95303	1350
Actual normal	3	96431

Table 4. Experimental results with dataset B

Algorithm		Accuracy %	Precision%	Recall %	FPR %	TNR %	FNR %	Training time (s)
Random Forest with the number of trees as	10	98.942	95.957	99.977	1.404	98.596	0.023	111.397
	40	98.943	95.965	99.973	1.401	98.599	0.027	127.143
	60	98.944	95.954	99.990	1.405	98.595	0.010	142.84
	80	98.944	95.958	99.987	1.403	98.597	0.013	156.939
	100	98.943	95.956	99.984	1.404	98.596	0.016	172.444
J48 [9, 21]		98.012	93.037	99.489	2.479	97.521	0.511	103.514
LSTM [21, 28]		96.579	94.801	91.313	1.667	98.333	8.687	201.2

Table 5. Confusion matrix result with the number of decision trees of 40 on dataset B

	Predicted as abnormal	Predicted as normal
Actual abnormal	96422	12
Actual normal	4067	285329

c. Experimental results with dataset C

When the number of abnormal records and the number of normal records have the largest difference, all experimental values give poorer results than other scenarios. This is reasonable because this is the nature of the classification process. If the disparity in the dataset is too large, the classification model will over fit. From Table 6 it can be seen that: with a parameter of the number of decision trees of 10, the highest Accuracy and Precision are respectively 99.016% and 94.825%. The confusion matrix values are shown in Table 7.

Table 6. Experimental results with dataset C

Algorithm		Accuracy %	Precision%	Recall %	FPR %	TNR %	FNR %	Training time (s)
Random Forest with the number of trees as	10	99.016	94.825	97.547	0.771	99.229	2.453	136.065
	40	98.877	92.254	99.473	1.209	98.791	0.527	168.027
	60	98.869	92.114	99.583	1.234	98.766	0.417	205.869
	80	98.786	91.262	99.971	1.386	98.614	0.029	243.109
	100	98.869	92.090	99.612	1.239	98.761	0.388	273.612
J48 [9, 21]		97.681	85.416	98.482	2.435	97.565	1.518	128.03
LSTM [21, 28]		94.752	88.939	90.209	3.735	96.265	9.791	400.642

Table 7. Confusion matrix result with the number of decision trees of 10 on dataset C

	Predicted as abnormal	Predicted as normal
Actual abnormal	94068	2366
Actual normal	5134	660839

d. Discussion

From the experimental results in Tables 2, 4 and 6, we can see that the RF algorithm gave good and stable classification results although there is a very large difference among the datasets. The lower the imbalance of the dataset is, the higher the measures of the correct detection rate are. Besides, for J48 [9, 21] and LSTM [21, 28] algorithms, when the dataset changes, the detection results and detection time also change. The J48 algorithm has the advantage of the lowest time for detection and classification due to using only one tree for evaluation. However, this algorithm has the disadvantage that its accuracy on all measurements is lower than the RF algorithm. With the LSTM algorithm [21, 28], the detection efficiency has been improved but the processing time is too slow compared with other algorithms. Thence it can see that the LSTM algorithm is not really suitable for datasets without time parameters. Based on these results, we provided some criteria and basis for cyber-attack detection systems to choose in order to balance between detection performance and time cost.

4.2. Evaluation of feature optimization methods

From Table 6, we select a parameter of the number of decision trees of 80 to conduct experiments and evaluate the feature optimization method. We chose this scenario because the dataset C and the number of decision trees of 80 give the lowest accuracy and precision.

4.2.1. Feature selection using correlation coefficient method

a. Experimental results of feature dimension reduction

According to the rule of selecting and evaluating features of the correlation coefficient method, if the two features have a large correlation coefficient, one of the two features should be removed. The reason is that if both features are kept, there is not mean much about terms of value. Accordingly, from Figure 1, we specify that if two features have the correlation coefficient greater than or equal to 0.9, or less than or equal to -0.9, one of the two features will be removed. By doing this, we removed 12 features consisting of *sloss, ct_state_ttl, synack, ct_dst_src_ltm, Dpkts, dwin, ackdat, ct_srv_dst, Ltime, dloss,* and *ct_src_ltm.* So from Figure 1, the number of remaining features is 31.

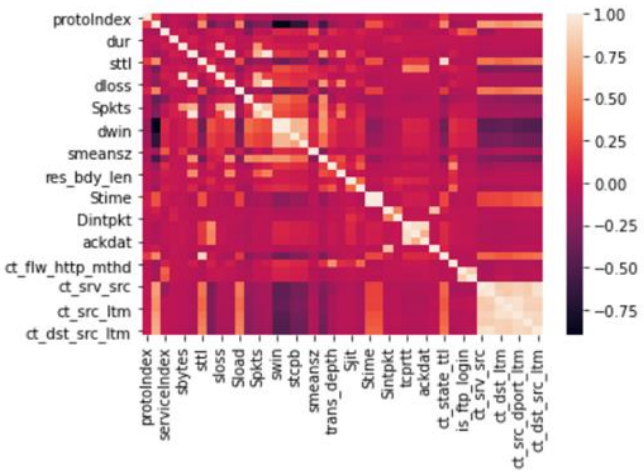

Figure 1. Correlation coefficient matrix among features in a dataset

b. Result of classification using correlation coefficient method

Experimental results of dataset C with 31 selected features are presented in Table 8. Comparing Table 8 with Table 6, we see that the important metrics such as accuracy, precision, and training time are all improved, being the following: Accuracy value increased by 0.015%; Precision value increased by 0.146%; Training time reduced by 52.954 seconds.

Table 8. Experimental results of dataset C with 31 features

Accuracy %	Precision %	Recall %	FPR %	TNR %	FNR %	Training Time (s)
98.801	91.408	99.945	1.365	98.635	0.055	190.155

4.2.2. Feature selection using IG method

a. Experimental results of feature dimension reduction

The Figure 2 shows the importance of each feature when using the IG evaluation method. Features with low importance scores (less than 0.01) will be removed to reduce the number of features. By doing this, we removed 15 features: *dloss, dwin, stcpb, dtcpb, trans_depth, res_bdy_len, Sjit, Djit, Stime, Ltime, is_sm_ips_ports, ct_flw_http_mthd, is_ftp_login, ct_ftp_cmd, ct_src_ltm.*

b. Result of classification using IG method

Experimental results of dataset C with 28 selected features are presented in Table 9. Comparing Table 9 with Tables 8 and 6, we see that the important metrics such as accuracy, precision, and training time are all much better, being the following: Accuracy value increased by 0.193%; Precision value increased by 2.333%; Training time reduced by 29.699 seconds.

Table 9. Experimental results of dataset C with 28 features

Accuracy %	Precision %	Recall %	FPR %	TNR %	FNR %	Training Time (s)
98.979%	93.595%	98.709%	0.982%	99.018	1.291	213.410

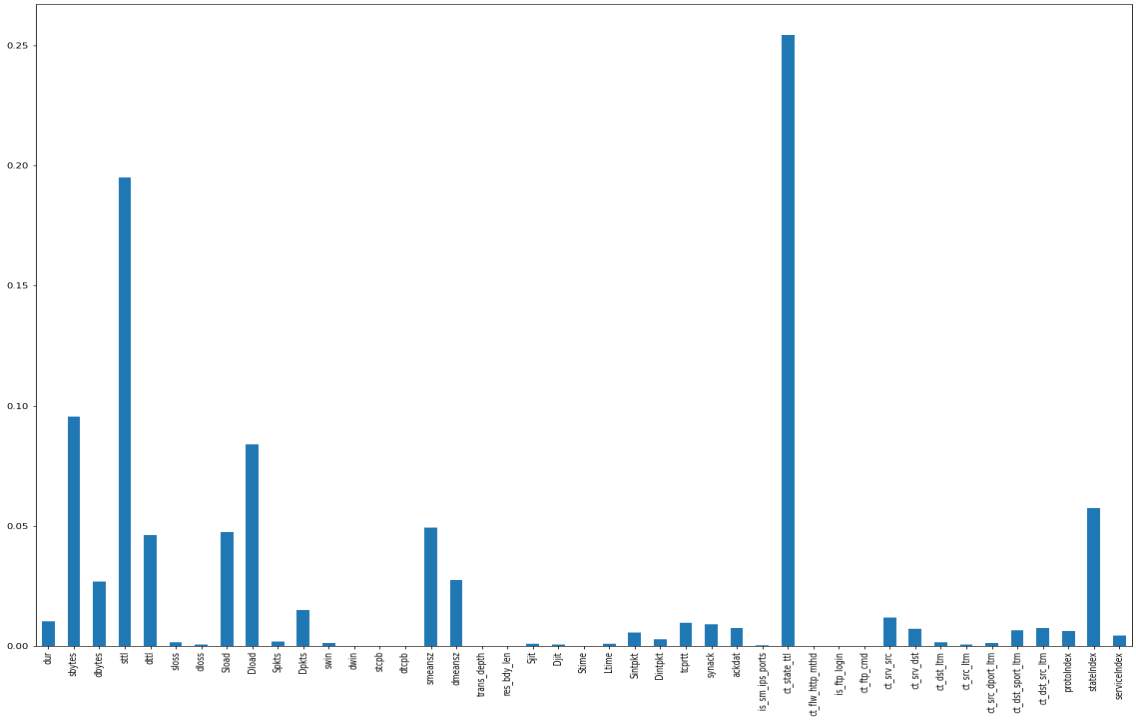

Figure 2. Graph of feature values by IG method

4.2.3. Feature selection using PCA method

a. Experimental results of feature dimension reduction

We choose to keep the number of features in dataset C at 31. After the experimental process, PCA method has removed 12 features consisting of Dpkts, dwin, ackdat, ct_srv_dst, Ltime, dloss, trans_depth, res_bdy_len, Sjit, Djit, Stime, Ltime, is_sm_ips_ports, and ct_flw_http_mthd.

b. Result of classification using PCA method

Experimental results of dataset C with 31 selected features are presented in Table 10. Comparing with the initial feature set, this experimental scenario also has better accuracy, precision, and training time values. Furthermore, reducing the feature dimension by PCA method has higher accuracy and precision than the feature selection using correlation coefficient method, but training time is more than 34.377 seconds. Comparing the experimental results in Table 10 with Table 9, the PCA method isn't as effective as the IG method. The reason is that the PCA method compresses data that could lead to the loss of important features, and the IG method performs weight evaluation to select features. Therefore, if the data set is larger, the use of the PCA method will be more effective.

Table 10. Experimental results of dataset C with 31 features

Accuracy %	Precision %	Recall %	FPR %	TNR %	FNR %	Training Time (s)
98.804%	91.926%	99.293%	1.267%	98.733%	0.707%	224.532

4.2.4. Discussion

The experimental results in Tables 8–10 show that the feature dimension reduction algorithms brought good efficiency in both 2 problems: improving the efficiency of the detection process, and time for detection and warning. However, based on the different efficiency of the feature dimension reduction methods, we noticed that cyber-attack monitoring and detection systems need a trade-off between detection efficiency and detection time. The IG and correlation coefficient algorithms can give better results in terms of detection time and efficiency if we continue to choose thresholds to reduce the dimension. However, if reducing the number of features too large, it will lead to the loss of data characteristics. Besides, these algorithms are only suitable for small and medium datasets. For large datasets, it is necessary to use the PCA method. Therefore, we think that monitoring systems need to constantly update and reevaluate the training model to change the values and roles of features to ensure that all useful features are used.

5. CONCLUSION

Cyber-attack techniques have always been and will always be major challenges for intrusion monitoring and detection systems. With the goal of optimizing the cyber-attack detection process, in our research, we proposed two main problems: optimizing the attack detection method by using the RF supervised learning algorithm and optimizing features based on feature dimension reduction techniques. The experimental results about detecting cyber-attacks using the RF algorithm show that the RF algorithm has been effective not only for the ability to accurately detect attacks but also for the ability to limit the false detection of attacks when the experimental dataset has a large difference between normal data and cyber-attack data. For the feature optimization process, feature dimension reduction methods removed many features. In particular, the correlation coefficient method decreased by 26%, IG decreased by 32%, and PCA decreased by 43% of the number of features. Although the number of features is reduced, the detection method still ensures the efficiency of accuracy as well as the detection time. This shows that dimensional reduction methods selected and eliminated accurately redundant features. With the results, our paper has not only provided network attack monitoring and detection systems with criteria to choose from to ensure the time and efficiency of the detection process but also proved that: to optimize the detection of cyber-attacks, it is not necessary to use advanced algorithms with complex and cumbersome computational requirements, it must depend on the monitoring data for selecting the reasonable feature extraction and optimization algorithm as well as the appropriate attack detection algorithms. In the future, we will continue to research and propose to apply our approach on other experimental data sets of cyber-attacks such as IDS 2018, CTU 13, etc. Besides, we will improve data dimension reduction solutions based on information representation methods of features or using graph theory.

REFERENCES

[1] R. Markus, *et al.*, "A survey of network-based intrusion detection data sets," *Computers & security*, vol. 8, no. 6, pp. 147-167, 2019.
[2] Kh. Ansam, *et al.*, "Survey of intrusion detection systems: techniques, datasets and challenges," *Cybersecurity*, vol. 20, pp. 2-20, 2019.
[3] A. T. Admassu, and S.N. Pramod., "A review on software defined network security risks and challenges," *TELKOMNIKA Telecommunication, Computing, Electronics and Control*, vol. 17, no. 6, pp. 3168-3174, 2019.
[4] Cyber Edge Group, "2019 Cyberthreat Defense Report," *Imperva*, 2019. [Online]. Available: https://www.imperva.com/resources/reports/CyberEdge-2019-CDR-Report-v1.1.pdf.
[5] Joe Levy, "Sophos 2020 Threat Report," *Sophos*, 2019, [Online]. Available: https://www.sophos.com/en-us/medialibrary/PDFs/technical-papers/sophoslabs-uncut-2020-threat-report.pdf.
[6] A. Mohiuddin, M. Abdun, H. Jiankun, "A Survey of Network Anomaly Detection Techniques," *Journal of Network and Computer Applications*, vol. 60. pp. 19-31, 2015.
[7] J. J. Arthur, *et al.*, "Review of the machine learning methods in the classification of phishing attack," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 8, vo. 4, pp. 1545-1555, 2019.
[8] D. X. Cho, *et al.* "An adaptive anomaly request detection framework based on dynamic web application profiles," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 5, pp. 5335-5346, 2020.
[9] The UNSW-NB15 Dataset Description, "University of New South Wales Canberra," 2020. [Online]. Available: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/.
[10] K. Vikash., *et al.*, "An integrated rule based intrusion detection system: analysis on UNSW-NB15 data set and the real time online dataset," *Cluster Computing*, vol. 23, pp. 1397-1418, 2019.

[11] N. Moustafa., *et al.,* "Novel Geometric Area Analysis Technique for Anomaly Detection using Trapezoidal Area Estimation on Large-scale Networks," *IEEE Transactions on Big Data*, vol. 5, no. 4, pp. 481-494, 2019.
[12] N. Moustafa *et al.*, "Big Data Analytics for Intrusion Detection System: Statistical Decision-Making Using Finite Dirichlet Mixture Models," *Chapter: 3Publisher: Springer publishing house*, 2017.
[13] S. Bagui, *et al.*, "Using machine learning techniques to identify rare cyber-attacks on the UNSW-NB15 dataset," *Security and Privacy*, vol. 2, no. 3, pp. 1-13, 2019.
[14] S. Rajagopal., *et al.*, "A predictive model for network intrusion detection using stacking approach," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2734-2741, 2020.
[15] S. Rajagopal, *et al.*, "Performance analysis of binary and multiclass models using azure machine learning," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 1, pp. 978-986, 2020.
[16] H. H. Ibrahim, *et al.*, "A comprehensive study of distributed Denial-of-Service attack with the detection techniques," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 4, pp. 3685-3694, 2020.
[17] M. Narender., and B.N. Yuvaraju, "Preemptive modelling towards classifying vulnerability of DDoS attack in SDN environment," *International Journal of Electrical and Computer Engineering (IJECE*), vol. 10, no. 2, pp. 1599-1611, 2020.
[18] J. Majidpour, and H. Hasan Zadeh, "Application of deep learning to enhance the accuracy of intrusion detection in modern computer networks," *Bulletin of Electrical Engineering and Informatics (BEEI)*, vol. 9, no. 3, pp. 1137-1148, 2020.
[19] N. Moustafa., and J. Slay., "The evaluation of Network Anomaly Detection Systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," I*nformation Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18-31, 2016.
[20] Z. M. Algelal, *et al.*, "Botnet detection using ensemble classifiers of network flow," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 2543-2550, 2020.
[21] P. Sun, *et al.*, "DL-IDS: Extracting Features Using CNN-LSTM Hybrid Network for Intrusion Detection System," *Security and Communication Networks*, vol. 2020, pp. 1-11, 2020.
[22] L. Breiman., "Understanding Random Forests: From Theory to Practice," *Machine Learning*, vol. 80, no. 1, pp. 5-32, 2017.
[23] S. S. Shai., "Understanding Machine Learning: From Theory to Algorithms," *Cambridge University Press*, 2018.
[24] B. Andrzej and J. Andrzej, "SPSS tutorial: Pearson Correlation," *Wydawnictwo Niezależne*. vol. 1, pp. 5-21, 2014.
[25] T. A. Alhaj, *et al.*, "Feature Selection Using Information Gain for Improved Structural-Based Alert Correlation," *PLOS ONE*, vol. 11, no. 11, 2016.
[26] J. Shlens., "A Tutorial on Principal Component Analysis," *arxiv.org/, arXiv:1404.1100*, 2014.
[27] J. Lever, Krzywinski, *et al.*, "Principal component analysis," *Nat Methods*, vol. 14, pp. 631-642, 2017.
[28] A. Boukhalfa, *et al.*, "LSTM deep learning method for network intrusion detection system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3315-3322, 2020.