Research on DDoS Attacks Detection Based on RDF-SVM

Chenguang Wang[1], Jing Zheng[2],Xiaoyong Li[1]

[1] Intelligent Traffic Data Security and Privacy Protection Lab, Beijing Jiaotong University, Beijing, 100044, China

[2] Shenzhen Medical Information Center, Shenzhen ,518000, China

*corresponding author's email: 15120368@bjtu.edu.cn

***Abstract*—DDoS attacks bring huge threaten to network, how to effectively detect DDoS is a hot topic of information security. Currently, there are some methods designed to detect DDoS attacks, but the detection rate of them is low. Moreover, DDoS detection is easily misled by flash crowd traffic. In this paper, a new method to detect DDoS attacks based on RDF-SVM algorithm is proposed. By considering the importance of feature selection in DDoS attacks detection, the RDF-SVM algorithm is designed to exploit random forest to compute the feature importance and SVM to rescreen the features, which will prevent from removing features mistakenly. Finally, an optimal feature subset is obtained, which will reach a higher detection rate and recall rate. In this paper, two kinds of datasets are used to train and test. The experimental result shows that the RDF-SVM algorithm can select the optimal feature subset over KDD99 dataset, and can also distinguish between DDoS attacks traffic and normal traffic (Flash Crowd) over the DDoS dataset collected from real environment. Compared with the CART, Neural Network, Logistic Regression, AdaBoost, and SVM method, RDF-SVM algorithm has a higher detection rate and recall rate.**

Keywords-DDoS, detection, RDF-SVM, feature selection, feature subset, Random Forest, SVM

I. INTRODUCTION

DoS (Denial of Service) makes the victim failed to receive legal requests, thus resulting in providing no services to legitimate users, especially DDoS (Distributed Denial of Service). DDoS attacks, generally divided into resource consumption and bandwidth consumption, control dozens of botnets to generate many forged packets. Hacker will implement DoS attacks on one or more targets and exhaust the victim's resources, so that the victim is incapable of providing normal network services[1].

With the development of network technology, DDoS attacks have been increasing year by year, severely disrupting the commercial operation, network environment and people's normal life. From 2002 to 2007, the global DNS root servers have repeatedly been attacked by DDoS, resulting in a large number of paralyzed servers; In March 2015, GitHub encountered large traffic DDOS attacks; In September 2016, hackers utilized Mirai to attack the Brian Krebs site and the attack traffic reached 665 Gbps[2, 3].

It is difficult for fake source IP DDoS to trace back. Moreover, the detection is easily misled by flash crowd traffic, which is very similar to the high-speed DDoS attacks. Accordingly, distinguishing between DDoS attacks and Flash Crowd is a research hotspot. However, there are some limitations for previous works to detect DDoS attacks, including low detection rate and high false positive rate. It is urgent to propose an approach to effectively detect DDoS attacks and filter the malicious traffic in backbones before they causes detriments to PC and server.

In this paper, a method based on RDF-SVM to identify and detect this attack is proposed, which combines the random forest and SVM. The approach utilizes the random forest to calculate variable importance and SVM algorithm to re-screen features. Finally, it will obtain the optimal feature subset with a higher detection rate. The innovation of this paper is following:

- Rescreen the features and prevent from deleting the features, which contribute to DDoS detection.
- Distinguish between DDoS attack traffic and Flash Crowd traffic;
- Suppress the attacks before it reaches the target host;
- Effectively filter out the random source IP attack;
- Detect known and unknown attack.

The outline of the paper is as follows. Section 2 reviews approaches related to this work. Section 3 introduces the random forest algorithm, SVM method and the RDF-SVM algorithm proposed by us. Section 4 analyzes the experimental result, and we present our conclusion in Section 5.

II. RELATED WORK

Recently, people have already done lots of work on detection. According to the comparison of algorithms, the detection can be mainly divided into three categories: based on source end, destination end and middle layer. We have come to the following conclusions [4]:

- Source end detection: It can effectively detect the forged source IP attacks, and is easy to trace back the attack host. but it would get a low detection rate, ISP can't benefit from it [5, 6].
- Destination end detection: ISP can gain benefit and it is easy to deploy. but a higher false rate, the host has been attacked before detected [7, 8].
- The middle layer detection: It can detect DDoS attacks before the attacks have happened. but many studies failed to be promoted to the backbone network [9, 10].

In conclusion, the above approaches have some limitations in the defense of DDoS attacks. Currently, algorithms of machine learning have been introduced into DDoS detection. Jiang Qi et al.[11] proposed a second detection and defense method using SVM, but the algorithm don't achieve pruning of the decision tree. Yan Wei et al.[12] proposed to transform the data nonlinearly and send the data to the self-organizing neural network. Saied A et

978-1-5386-1230-9/17 $31.00 © 2017 IEEE
DOI 10.1109/ICICTA.2017.43

al.[13] only analyzed common protocols used by DDoS attacks, and updated their databases timely. But, the approach can't handle encrypted packages. Li Chunlin et al.[14] used the self-coding network model to extract network features. The softmax classifier was used to classify the data without label. However, the detection rate is not optimal. Kouguang et al.[15] transformed the 20 features into grayscale images, image recognition never deliberately conceals features. However, DDoS attacks conceal their characteristics, so the algorithm can't reach the best precision rate.

Feature selection is an important step in machine learning, so extracting the optimal feature subset contributes to accelerate the accuracy of detection. Yao Dengju et al.[16] proposed a RDFFS algorithm, which used sequence backward selection and generalized sequence backward selection method to select features. Jiang Shengyi et al.[17] sorted the features based on size of the clusters to obtain the feature subset. However, its detection rate was low. Htun PT et al.[18] applied the combination of random forest and KNN algorithm to select features and detect DoS attacks. But, this approach will mistakenly remove features contributing to the classification positively.

These approaches fail to meet several major requirements, which should include high detection rate and high recall rate. In this paper, we design an algorithm based on RDF-SVM, which exploits random forest to compute the features importance and SVM to rescreen the features. They can effectively prevent from removing the features contributing to classify and eventually obtain the optimal feature subset.

III. PROPOSED ALGORITHM

A. Random Forest

The random forest algorithm is an integrated machine learning, which introduces random sampling technique (bootstrap) and the node-splitting technique to construct multiple decision trees randomly, and gains the final classification result by voting.

Random forest is made up of several independent classification models $\{ (\mathrm{X}, \theta_K) | \, k = 1,2 \ldots \ldots \}$ of decision tree, in which $\{ \theta_K \}$ is assumed to be independent and identically distributed random variable, K represents the number of random decision trees. Its structure is as follows:

- Select *N* samples randomly and put back after selected every time, in order to train decision tree.
- Each sample has *M* attributes, when the decision tree node is split. We randomly select m $(\mathrm{m} \ll \mathrm{M})$ attributes, and use the information gain strategy to select an attribute as the split attribute of the node.
- Each node in the decision tree is split in step 2 until it can't be split again.
- Build amounts of decision trees according to steps 1 to 3 without pruning.
- The decision trees vote on the input variables X_i and count them, then the highest votes is the label of classification.

In this paper, we choose ID3 as the algorithm of computing feature importance, which bases on the information gain.

Entropy represents a disorder state. The information entropy is defined as:

$$\mathrm{Entropy}(x) = \textstyle\sum_{i=1}^{c} -P_i \log_2 P_i \qquad (1)$$

The random variable *X* is a finite random discrete variable, P_i is the ratio of the random variable x_i and the dataset *C*. If the training sample *x* is divided by feature *Y*, the expectation of *Y* for *x* is:

$$\mathrm{Entropy}_Y(x) = \textstyle\sum_{j=1}^{N} \frac{|x_i|}{|x|} \mathrm{Entropy}_Y(x) \qquad (2)$$

When the uncertainty of training samples rises, the information entropy also increases. The definition of information gain is as follows:

$$\mathrm{Gain}(x, Y) = \mathrm{Entropy}(x) - \mathrm{Entropy}_Y(x) \qquad (3)$$

Entropy is used as a measure of information gain. The greater the information gain is, the more important the feature is. Therefore, the feature makes contributions to the detection rate.

B. SVM algorithm

SVM is 2-classification algorithm, which performs well in generalization and handling with imbalanced dataset. Accordingly, we select it for rescreening feature.

For the training set $\mathrm{D} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$, $y_i \in \{-1,1\}$, the basic notion of classification learning is to find a hyper-plane among the samples. The hyper-plane is represented by the linear equation:

$$\omega^T x + b = 0 \qquad (4)$$

ω is normal vector set, *b* is displacement. The distance *r* of any point *x* of sample space to hyper-plane (ω, b) is expressed as follows:

$$r = \frac{|\, \omega^T x + b|}{||\omega||} \qquad (5)$$

Suppose that the hyper-plane could train samples correctly. Namely, for $(x_i, y_i) \in D$, then $y_i = +1$, so $\omega^T x + b > 0$. If $y_i = -1$, then $\omega^T x + b < 0$.

Finally, the most suitable variables ω and *b* will be obtained, which receives the largest *r*. Therefore, the optimization problem of SVM is as followed :

$$\max_{\omega, b} \frac{2}{||\omega||} \qquad (6)$$

$$\mathrm{s.t.} \;\; y_i(\, \omega^T x_i + b) \geq 1, \; i = 1,2, \ldots \ldots, m \qquad (7)$$

The *Lagrange* optimization method make the optimal hyper-plane problem dual, namely, if it satisfy $\sum_{i=1}^{n} a_i y_i = 0 \; and \; a_i \geq 0, (i = 1, \ldots, n)$, the optimum solution

$Q(a)\ is\ shown\ as\ follows$:

$$Q(a) = \sum_{i=1}^{n} a_i - \frac{1}{2}\sum_{i,j=1}^{n} a_i a_j y_i y_j (x_i \cdot x_j) \qquad (8)$$

At last, the method would receive optimal classification function $f(x)$:

$$f(x) = sgn\{\sum_{svm} a_i^{\cdot} y_i\, (x_i \cdot x) - b^{\cdot}\} \qquad (9)$$

C. *Proposed RDF-SVM algorithm*

The RDF-SVM algorithm is designed to detect DDoS attacks. The feature importance is calculated by Random Forest. Sorting the features $\{ x_i\ |i = 1,2,3 \ldots\ldots, n\}$ firstly is to obtain its importance set *W*. However, the feature contributing to the classification may be removed mistakenly, so we rescreen the features by SVM.

If the feature promotes to classify, we define it as a positive feature. The rest of them are negative features. The contribution function is shown as $F(x_i)$:

$$F(x_i) = \begin{cases} 1, & x_i\ is\ positive\ feature \\ -1, & x_i\ is\ negative\ feature \end{cases} \qquad (10)$$

We set a threshold α for ω_{x_i}. If $\omega_{x_i} < α$, then it would be used for training and predicting with deleting (the precision rate is $P_j^{(1)}$) and not deleting (the precision rate is $P_j^{(2)}$) it by SVM method separately, the value sum_P represents the difference (shown in line 17):

$$\text{sum\_P} = \text{P}_j^{(1)} - \text{P}_j^{(2)} \qquad (11)$$

Then *sum_P* will be received for n times and finally get the average value Mean_sumpre_diff (shown in line 20).

$$\text{Mean\_sumpre\_diff} = (\textstyle\sum_j(\text{P}_j^{(1)} - \text{P}_j^{(2)}))\ /\ n, j = 1,2,\ldots \qquad (12)$$

If Mean_sumpre_diff > t, then $x_i \in C$, the feature will be retained. Otherwise, $x_i \notin C$, then it will be removed. Repeating the above steps until all features ($\omega_{x_i} < α$) are analyzed. Set C will eventually be updated. The algorithm is shown in the following table 1.

Table 1 RDF-SVM Algorithm

```
Algorithm 1   RDF-SVM algorithm
Input:  $\{ x_i| i = 1,2,3 \ldots\ldots, N\}, S_{train}, S_{test}$
Output:  Optimal feature subset C
Steps:
1:  initial ω ← ϕ, sum_P ← 0, Mean_sumpre_diff ← 0
2:  for each $x_i$ do
3:        $\omega_{x_i}$ ← Randomforest($x_i$);
4:  end for
5:        W ← Sort($\omega_{x_i}$);
6:  for each $\omega_{x_i} \in W$ do
7:        if $\omega_{x_i} > \alpha$ then
8:              A_index ← $x_i$;
9:        else
10:             B_index ← $x_i$;
11:       end for
12: end for
13: for each $X_i$ ∈ B_index do
14:    for $j \in (1, n)$  do
15:          $P_j^{(1)} = SVM$(y$_i$, $x_i$, y_train, X_test);
16:          $P_j^{(2)} = SVM$(y$_i$, A$_{x_i}$, y_train, A_ X_test);
17:          $sum\_P = sum\_P + P_j^{(1)} - P_j^{(2)}$;
18:          j++;
19:    end for
20:          Mean_sumpre_diff  = sum_P / n;
21:     if Mean_sumpre_diff > t then
22:           A_index ← $x_i$;
23:      else
24:            remove($x_i$);
25: end for
26: return A_index;
```

The algorithm is implemented in python language, which calls Randomforest and SVM library. After the algorithm has been trained, the number of decision trees n_tree is 10, after trained, ω takes the value of 0.01, Mean_sumpre_diff is 0.000000001, and n is 20.

IV. Experimental results and analysis

In this section, we carried out experiments on the KDD99 dataset and the DDoS dataset collected from the testing environment. Then we summarize our experimental results, evaluate our approach and analyze the results.

A. *Experimental analysis of KDD99 dataset*

The KDD99 database contains 42 features of normal and attack dataset. We choose the dataset KDD Train+ and KDD Test+ as the training set and testing set. KDD Test+ contains of 39 kinds of attacks, which is 17 kinds more than KDD Train+. Therefore, this approach selects KDD Train+ as training set and KDD Test+ as test set, so that it will validate the precision rate of unknown attacks detection. In the experiment, the ratio of training set to testing set is 4:1.

To make sure the RDF-SVM algorithm can obtain more optimal feature subset and receive a better detection rate. We made a comparison among three methods. The first method used all features to classify by SVM. The second method used the feature subset, which removes the low weight features directly, to classify by SVM. The third method is the RDF-SVM algorithm. The feature subsets of three methods are shown in the table 2.

Table 2 Three Methods' Feature Subset

	Features
SVM	42
RF and SVM	17
RDF-SVM	30

The paper compares precision rate and recall rates of the three methods, as shown in Figure 1 and Figure 2. From the below figures, we can see the RDF-SVM algorithm with 82.85% precision rate and 80.09% recall rate. Although the feature subset obtained by RDF-SVM algorithm is not minimal, its precision rate and recall rate are the highest. To conclude, RDF-SVM algorithm can prevent from removing the feature subset, which promotes to classification.

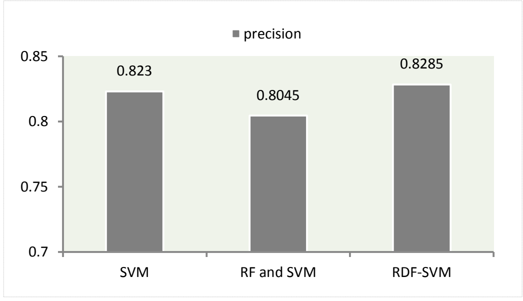

Figure 1. Precision Rate Among Three Methods

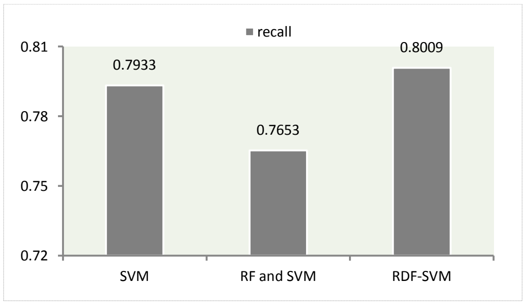

Figure 2. Recall Rate Among Three Methods

In order to future validate performance of the algorithm, this paper compares the method with several other machine learning methods, including CART, Neural Network, Logistic regression, AdaBoost and SVM. The results are shown in Figure 3 and Figure 4.

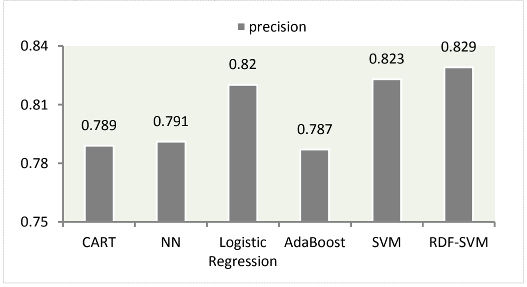

Figure 3. Comparison Of Different Methods Over Precision Rate

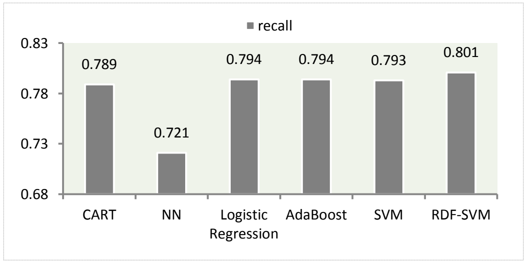

Figure 4. Comparison Of Different Methods Over Recall Rate

We can draw conclusions: the precision rate and recall rate of RDF-SVM algorithm are the highest and RDF-SVM algorithm can detect known and unknown attacks effectively.

B. *Experimental analysis of DDoS dataset*

Experiment environment (shown in Figure 5) is built in the LAN. Experimental equipment contains a Dell Server as a f server, another server as a management server, two TP-LINK TL-SG1016DT switches, two thin clients as the attacked end, and five thin clients as the attack end.

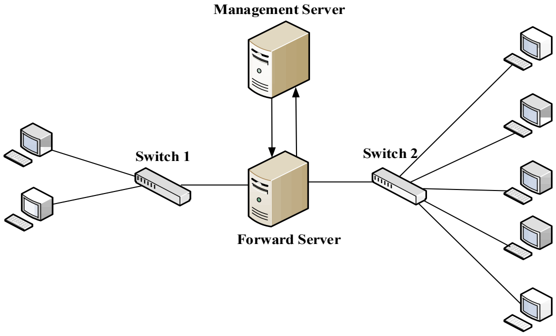

Figure 5. Network Topology

Gateway server, based on the Netfilter/iptables framework, intercepts all forwarded packets and sends them to the Management server. The management server will receive and analyze the packets, eventually discard the random IP which only occur once. If the source IP, destination IP, source port, destination port, packet size and protocol type of the packets precisely match, we will count them as a flow and statistic flow information into the database. A flow contains 128 time fragments, and each fragment corresponds to the number of packets.

The experiment mainly collected three types of data: random IP DDoS, real IP DDoS and Flash crowd. The random source IP DDoS data is generated by TFN2K. Although the model has discarded packets with random source IP, the database will record the pseudo-random flow generated by TFN2K, due to the same random IP. Trinoo generats the real source IP DDoS traffic. Flash crowd is received by constantly accessing web pages and videos. The three kinds of flow are show in figure 6:

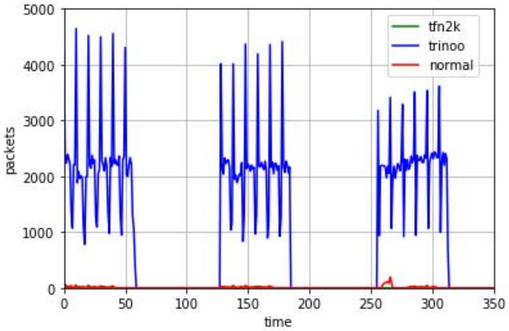

Figure 6. The Flow Of Flows TFN2K, Trinoo And Normal

To analyze the characteristic of flows, 14 basic flow features are extracted to detect DDoS. (e.g. the probability of source IP address, probability of destination IP address, length of each packet, protocol type, total packet byte, the average packet byte, variance of packet byte, standard deviation of packet byte, average of bandwidth, average of packet number, variance of packet number, standard deviation of packet number, the non-zero fragment of flow, the maximum fragment of flow, and the number of packets in first time fragment). Training data and testing data are shown in the following table 3:

Table 3 Numbers Of Training And Testing Set

	Training set	Testing set
TFN2K	2000	500
Trinoo	2000	500
Flash Crowd	500	125

After trained and tested by RDF-SVM algorithm, the feature subset {1,0,3,10,11,8,12,13,9,6,2} is obtained, and the method of RF and SVM removed mistakenly 2-th feature. Then we make a comparison RDF-SVM algorithm with other machine learning methods over the DDoS dataset, the result is shown in Table 4.

Table 4 Comparison With Other Approaches

	Precision	Recall	F-score
CART	0.9825	0.9818	0.9794
Neural Network	0.9058	0.9444	0.9245
Logistic Regression	0.9632	0.9291	0.9409
AdaBoost	0.9821	0.9684	0.9722
SVM	0.9851	0.9856	0.9852
RDF-SVM	0.9872	0.9875	0.9869

Experimental results show that the precision of RDF-SVM algorithm in detecting DDoS attacks is 98.72%, the recall rate is 98.75%, and the f-score is 98.69%, whose performance is better than other approaches. Meanwhile, we can conclude that it can effectively distinguish random IP address attacks, real IP address attacks and Flash Crowd.

V. Conclusion

In this paper, we have analyzed the characteristics of DDoS attacks, and proposed a novel detection model to detect DDoS attacks using RDF-SVM algorithm, which extracted from DDoS flows, utilize SVM to rescreen the features and obtain the optimal feature subset finally. This algorithm can compute the important features well, and the experimental results show that the algorithm can get better classification performance and more optimal feature subset, Besides, which can detect known and unknown attacks and distinguish random IP address attacks, real IP address attacks and Flash crowd more effectively compared with other methods.

Acknowledgment

This work was supported by the National Basic Research Program Research on application technology of Railway Information Service Program (Grant No. 2016X004-C), Study on Safety Assessment Guide for Key Network and Important Information System in Transportation Industry (Grant No. K15I00030).

References

[1] Yan Fen, Wang Jiajia, et al. "Summary of DDoS attack detection." Computer application research, 2008, vol.25, no.4, PP.966-969.

[2] http://zt.360.cn/1101061855.php?dtid=1101062514&did=1101843681. 2016.

[3] http://www.h3c.com/cn/About_H3C/Company_Publication/IP_Lh/2013/06/Home/Catalog/201401/812468_30008_0.htm. 2013.

[4] Zhang Yunzheng and Xiao Jun, "DDoS Attack Detection and Control Methods." Journal of Software, 2012, vol.23, no.8, pp.2058-2072.

[5] Petiz, I., P. Salvador, A. Nogueira, and E. Rocha, "Detecting DDoS attacks at the source using multiscaling analysis." Telecommunications Network Strategy and Planning Symposium, 2014, vol.16, pp.1-5.

[6] Williamson, M.M, "Throttling viruses: restricting propagation to defeat malicious mobile code." Computer Security Applications Conference, 2002. Proceeding, 2009, pp.61-68.

[7] Chen Chen, and Xu Yang, "Research on DDoS based distributed defense system based on SDN." Fujian computer, 2017, vol.33, no.4, PP.13-16.

[8] Liu Anli, and Zhao Huaixun, "a fast algorithm for DDoS attack source tracking based on traffic intensity." Modern electronic technology, 2010, vol. 33, no.7, PP.131-134.

[9] Yang Jungang, Wang Xintong, and Liu Guqing, "DDoS attack detection methods based on traffic and IP entropy." Computer application research, 2016, vol.33, no.4, PP.1145-1149.

[10] Zhao Xin, and Zhang Yu, "Defensive strategies for adaptive DDoS attacks based on border gateways." Intelligent computers and applications, 2011, vol.01, no.3, PP.79-82.

[11] Jiang Qi, Zhuang Yi, and Xie Dong, "Research on generation method of SYN Flood attack detection rules based on SVM classifier." Computer applications and software, 2005, vol.22, no.10, PP.38-39.

[12] Yan Wei, and Nanyang, "An improved network intrusion detection algorithm based on SVM." Science and Technology Bulletin, 2012, vol.28, no.10, PP.158-159.

[13] Saied, A., R.E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using Artificial Neural Networks." Neurocomputing, 2016, vol.172, PP.385-393.

[14] Li Chunlin, Huang Yuejiang, Wang Hong, and Niu Changxi, "A network intrusion detection method based on deep learning." Information security and communication security, 2014, vol.04, no.10, PP.68-71.

[15] Kou Guang, Tang Guangming, Wang Shuo, Song Haitao, and Bian yuan, "Depth study on the application of cloud detection in zombies." Journal of communications, 2016, vol.37, no.11, PP.114-128.

[16] Yao Dengju, Yang Jing, and Zhan Xiaojuan, "A feature selection algorithm based on random forests." Journal of Jilin University: Engineering Science, 2014, vol.44, no.1, PP.137-141.

[17] Jiang Shengyi, Zheng Qi, and Zhang Qiansheng, "A feature selection method based on clustering." Journal of electronics, 2008, vol.36, no.12, PP.157-160.

[18] T, H.P., K.O.K. T, "Detection Model for Daniel-of-Service Attacks using Random Forest and k-Nearest Neighbors." International Journal of Advanced Research in Computer Engineering & Technology, 2013, vol.02, no.2, PP.1855-1860.

[19] Li Xiaoming, Ren Hui, and Yan Jinyao, "Analysis and research of network traffic classification algorithm based on machine learning." Journal of Communication University of China, 2017, vol.24, no.2, PP.9-14.

[20] Zhao Wei, "Comparison of feature selection methods using machine learning clustering model." Journal of Huaqiao University, 2017, vol.38, no.1, PP.105-108.