Future Generation Computer Systems 106 (2020) 347–359

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

MF-Adaboost: LDoS attack detection based on multi-features and improved Adaboost

Dan Tang [a], Liu Tang [a], Rui Dai [a], Jingwen Chen [a], Xiong Li [b], Joel J.P.C. Rodrigues [c,d,*]

[a] College of Computer Science and Electronic Engineering, Hunan University, Changsha, 410082, China
[b] Institute for Cyber Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[c] Federal University of Piauí, Teresina, PI, Brazil
[d] Instituto de Telecomunicações, Portugal

ARTICLE INFO

Article history:
Received 18 April 2019
Received in revised form 13 December 2019
Accepted 25 December 2019
Available online 2 January 2020

Keywords:
LDoS
Internet of Things
Feature set
Feature selection
Adaboost algorithm
Machine learning

ABSTRACT

A low-rate denial of service (LDoS) attack is a precise network attack that aims at reducing the quality of the network service. Many networks do not have an effective mechanism for defending against LDoS attacks, including the emerging Internet of Things. In this paper, we propose an LDoS attack detection method that is based on multiple features of network traffic and an improved Adaboost algorithm (MF-Adaboost). Based on an analysis of the network traffic, we construct a network feature set that is used for feature calculation and feature selection of network traffic data. Feature calculation can extract the most useful information from the network traffic data and reduce the scale of the network data. Feature selection is used to select the optimal classification features to ensure that the detection algorithm can be effectively trained. This method utilizes the Adaboost algorithm, which is a classification algorithm in the field of machine learning. The well-trained Adaboost algorithm can effectively identify LDoS attack traffic. We improve the Adaboost algorithm to alleviate the imbalance of the sample weights. Experiments are conducted on the NS2 simulation platform and a test-bed platform to evaluate the performance of our method. The experimental results demonstrate that our method can detect LDoS attacks effectively.

1. Introduction

A low-rate denial of service (LDoS) attack is a variant of denial of service (DoS) attack. A DoS attack is an early type of network attack. It uses a botnet to send data packets to the network via a flooding approach, which paralyzes the whole network and consumes a substantial amount of network resources. However, with the development of network security technology, many network devices can quickly identify DoS attacks and take corresponding security measures to defend against it. Therefore, to avoid being detected and to improve the attack efficiency, many variants of DoS attacks have been developed, such as LDoS attacks. In contrast to DoS attacks, an LDoS attack is a precision attack. It exploits the vulnerabilities in network protocols to launch attacks and often realizes superior attack effects at a smaller attack cost [1]. Because the pulses that are emitted by an LDoS attack are periodic rather than persistent, the average rate of an LDoS attack is very low. Hence, an LDoS attack has superior concealment and a variable attack approach, which hinders detection by current network security mechanisms. Many network platforms, such as cloud computing platforms [2], the Internet of Things (IoT) [3–5], wireless sensors network(WSN) [6–11] and the software-defined network (SDN) [12], are also facing the threat of LDoS attacks.

Therefore, we propose a method that is based on multiple features of network traffic and an improved Adaboost algorithm (MF-Adaboost) for detecting LDoS attacks in network traffic. Network traffic is a kind of time series; hence, time series methods can be used to analyze changes in the network traffic [13,14]. A time series has many statistical characteristics that can be used to represent the state of the network [15,16]. If an attack occurs, these characteristics will change to varying degrees. The impacts of LDoS attacks on the network traffic characteristics is differ among networks due to differences in the network environments, protocols and topologies. Therefore, it is necessary to analyze the correlations between the changes in the network traffic internal characteristics and the occurrence of LDoS attacks, which can be used as a basis for detecting LDoS attacks. In the proposed method, we use the Chi-squared test algorithm to analyze this correlation.

* Corresponding author at: Federal University of Piauí, Teresina, PI, Brazil.
E-mail addresses: tangliu@hnu.edu.cn (L. Tang), lixiong@uestc.edu.cn (X. Li), joeljr@ieee.org (J.J.P.C. Rodrigues).

https://doi.org/10.1016/j.future.2019.12.034
0167-739X/

We use the improved Adaboost algorithm as the classifier to classify the network traffic based on network traffic characteristics. An LDoS attack changes the statistical distribution of the network traffic eigenvalues. Therefore, the preattack network traffic and the postattack network traffic can be regarded as two types of network traffic. The classifier can learn and remember the characteristics of these two kinds of network traffic from a training set. With multiple trainings, the classifier will be able to identify LDoS attacks. The main contributions of this paper are as follows:

- A new LDoS attack detection method is proposed that uses the Adaboost algorithm to classify the network traffic [17]. By training the Adaboost classifier, the data that contain an LDoS attack can be detected in the network.
- An improved Adaboost algorithm is developed. With the objective of addressing the imbalance of the sample weight distribution in the traditional Adaboost algorithm, an improved algorithm for updating the sample weights is proposed.
- This method can adaptively adjust the parameters of the algorithm during the training process. Therefore, the method can adapt to the complex and varied network environment by regularly updating the training data.
- This method has demonstrated satisfactory detection performance in experiments on the NS2 simulation platform and a test-bed platform.

The remainder of this paper is organized as follows: Section 2 discusses various LDoS attack methods. Section 3 describes the feature set that we built. Section 4 describes the principle and strategy of the LDoS attack detection method that is based on MF-Adaboost in detail. The experiments will be explained in Section 5. In this section, we use the NS2 simulation experimental platform and a test-bed experimental platform to evaluate the performance of our method. The conclusions of this study will be presented and future work will be discussed in Section 6.

2. Related work

Many methods study network anomalies in both the time and frequency domains. Therefore, many signal processing techniques are used to model and detect LDoS attack traffic. By combining power spectrum analysis with information entropy, Chen et al. proposed two new information metrics for detecting LDoS attacks: the Fourier power spectral entropy (FPSE) and the wavelet power spectral entropy (WPSE) [18]. Since the energy of an LDoS attack signal is mainly concentrated in the low-frequency range, its FPSE and WPSE are lower than those of normal traffic. Based on these two indicators, a robust RED queuing algorithm that is based on the power spectral entropy is proposed for mitigating LDoS attacks. Agrawal et al. proposed a method for detecting and mitigating LDDoS attacks in the frequency domain [19]. The method is based on the power spectral density (PSD), which enables real-time monitoring and analysis of the network traffic for detecting LDDoS attacks. It consists of five phases. The first to fourth phases are in the time domain and the fifth phase is in the frequency domain. In addition, it can adaptively detect the internal and external traffic of a cloud network. Wu et al. proposed an LDoS attack stream filtering method that was based on spectrum analysis [20]. This method converts the TCP traffic and the LDoS attack flow from the time domain to the frequency domain. Then, a frequency-domain search method is used to estimate the round-trip time (RTT). According to an analysis of the frequency spectrum, the energy of the TCP traffic is mainly distributed among the N/RTT points. Therefore, a comb filter can be used to filter LDoS attack streams in the frequency domain. The filter allows the energy of most legitimate TCP traffic at the N/RTT points to pass through. Wu et al. proposed a correlation-based LDoS attack detection method [21]. This method does not directly calculate the correlation coefficient of the network traffic sequence but calculates the correlation coefficient of the Hilbert spectrum of the network traffic. The Hilbert spectrum that is obtained via the Hilbert–Huang transform is an energy–frequency–time distribution that provides more network traffic information than the original network traffic sequence.

Various methods transform anomaly detection into traffic classification by analyzing the changes in network features under normal and abnormal conditions. Aiming at identifying the multifractal characteristics of network traffic, Yue et al. proposed a network traffic identification method that is based on the wavelet transform and a neural network [22]. This method extracts the wavelet energy spectrum coefficients of the network traffic and uses them to analyze the multifractal characteristics of networks at various time scales. The wavelet energy spectrum coefficients are differ between normal traffic and abnormal traffic. Hence, this method utilizes a neural network to classify the network traffic by using wavelet energy spectrum coefficients. Yi et al. proposed a detection method that was based on a traffic classification model for a new BGP-LDoS network attack [23]. First, three network features are selected via statistical analysis. Then, based on conditional random field theory, a traffic classification model is established, which can classify unlabeled network traffic. Finally, the classification results of the model are used to determine whether an LDoS attack has occurred. Xie et al. proposed accurate network traffic anomaly detection by tensor factorization [24–27] and parameters compressed NN [28] which cannot be used to detect LDOS attack directly.

The traditional classification method need to design handcrafted feature or adopt statistical features, which requires a considerable amount of engineering skill and domain expertise. Good features [29] can be learned automatically using deep learning. As the most common model of deep learning, convolutional neural networks (CNNs) have achieved great success in the detection [30], segmentation, classification [31,32] and tracking [33] of objects in images. Images or audio spectrograms are 2D arrays, and video or volumetric images are 3D arrays. However, sequences such as language, speech and network traffic, are 1D arrays. So the traditional way of "feature extraction + classifier" may be more suitable for network traffic classification.

A network is a kind of time series. Many people study LDoS attack detection methods from the perspective of sequences. These methods must always set a detection threshold, which is used to determine the occurrence of an LDoS attack. Wu et al. proposed a distributed LDoS attack detection method that was based on sequence matching [34]. This method studies the characteristics of LDoS attacks from the perspective of sequence matching, and finds that the LDoS attack pulse sequence that is received by the victim has the characteristic of similarity. Therefore, they propose the use of sequence alignment detection techniques in bioinformatics to detect LDoS attack flows. This method uses the Smith–Waterman local sequence alignment algorithm to obtain the similarity score of two sequences. This method can estimate the parameters of an LDoS attack accurately based on the pulse periods, lengths and amplitudes of the detection sequences. Bhuyan et al. proposed a mechanism for detecting multiscale LDDoS attacks by using a generalized total variation metric [35]. This metric is sensitive to changes that occur in the network, and its values differ substantially between legitimate traffic and attack traffic. Tang et al. proposed an LDoS attack detection method that is based on two-step clustering [36]. This method uses a two-step clustering algorithm to cluster network traffic and detects LDoS attacks by analyzing anomaly clusters.

Table 1
Features of network traffic.

Total TCP	Total UDP
Total data	TCP ratio
UDP ratio	TCP average
UDP average	TCP variance
UDP variance	TCP and UDP covariance
TCP information entropy	UDP information entropy
TCP Hurst value	UDP Hurst value
TCP maximum	UDP maximum
TCP minimum	UDP minimum
TCP range	UDP range
TCP skewness	UDP skewness
TCP kurtosis	UDP kurtosis
TCP variation coefficient	UDP variation coefficient
TCP mean absolute difference	UDP mean absolute difference

Through the above analysis, we can conclude that available methods have the following shortcomings:

- High false-positive rate and false-negative rate: The methods that involve the frequency domain and the time domain cannot accurately distinguish between normal flows and LDoS attack flows. Most have higher false-positive and false-negative rates.
- Poor adaptability: The methods with a detection threshold cannot adjust the threshold adaptively to cope with changes in the environment.

To address these limitations, the proposed method combines multiple characteristics of network traffic and the improved Adaboost algorithm to detect LDoS attacks in the network. The combination of features can more accurately reflect network anomalies and the Adaboost algorithm is a stable and adaptive classification algorithm in the field of machine learning. Therefore, our detection model has satisfactory adaptability and can effectively detect LDoS attacks in the network.

3. Feature set

The network traffic is sampled at fixed intervals, and the obtained sequence is a kind of time series [37]. Mathematically, many statistics can be used to reflect changes in time series. These statistics include: the total amount of network data, the proportion of each component, the mean and variance of each component, the maximum and minimum of each component, the correlation coefficient, the skewness, the kurtosis, and the range, among others. In addition to these statistics, we use other features to detect LDoS attacks, such as the Hurst value and the information entropy. The features that are used in this paper are listed in Table 1. The 28 features in the table constitute a feature set.

In Table 1, TCP refers to the TCP traffic sequence and UDP refers to the UDP traffic sequence. For a network traffic sequence over a period of time, Total TCP represents the total number of packets in the TCP traffic sequence, Total UDP represents the total number of packets in the UDP traffic sequence, Total data represents the total number of packets in the network traffic sequence, TCP ratio represents the proportion of the TCP traffic relative to the total network traffic, UDP ratio represents the proportion of the UDP traffic relative to the total network traffic, and TCP and UDP covariance represents the covariance of the TCP traffic sequence and the UDP traffic sequence. An LDoS attack causes constant network congestion by periodically generating high pulses. Under the attack of LDoS, the TCP traffic will be in a fluctuating and unstable state due to the congestion control protocol, and the UDP traffic will also change over a small time scale. Therefore, the time series of the TCP traffic and the time series of the UDP traffic will be abnormal. These anomalies can be reflected in the statistical changes of the time series, such as the TCP variance and the UDP variance. As shown in Fig. 1, the LDoS attack causes the TCP variance and the UDP variance to fluctuate.

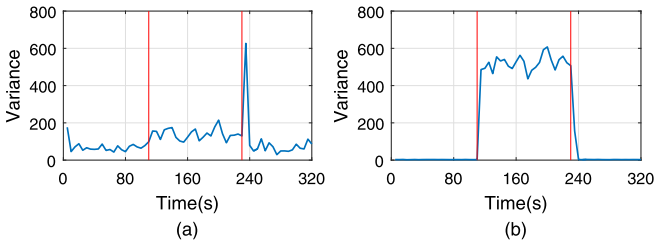

Fig. 1. (a) The TCP variance and (b) the UDP variance.

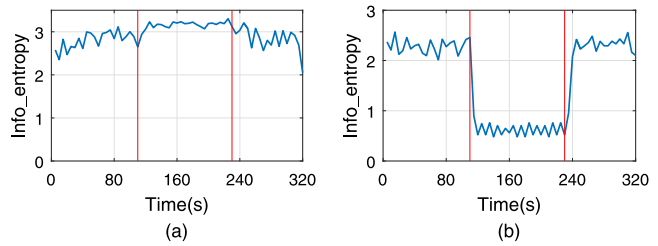

Fig. 2. (a) The TCP information entropy and (b) the UDP information entropy.

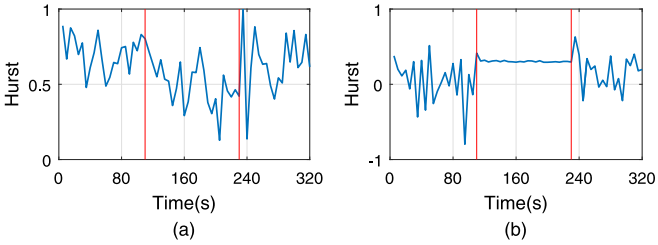

Fig. 3. (a) The TCP Hurst value and (b) the UDP Hurst value.

In mathematical theory, the information entropy is used to represent the uncertainty of information. For a time series, the information entropy represents the degree of disorder of the sequence. The larger the value of the information entropy is, the more disordered the sequence is, and the smaller the value of the information entropy is, the more stable the sequence is. The impacts of the LDoS attack on the information entropies of the TCP traffic and the UDP traffic are shown in Fig. 2.

Network traffic has the characteristics of self-similarity and long correlation [38], and the Hurst exponent is an important parameter for characterizing the burstiness of network traffic. The high pulse of the LDoS attack can cause sudden changes in the TCP traffic and the UDP traffic. Hence, the Hurst values of the TCP traffic and the UDP traffic will change. Fig. 3 shows the impacts of the LDoS attack on the TCP Hurst value and the UDP Hurst value.

4. Detection method that is based on MF-Adaboost

In this section, we will introduce our method in detail.

4.1. Model of the method

Fig. 4 illustrates the model of our method. In this model, the network traffic is divided into a training set and a testing set. The data of the training set are used to train the original classifier. Therefore, the training set must contain attack data

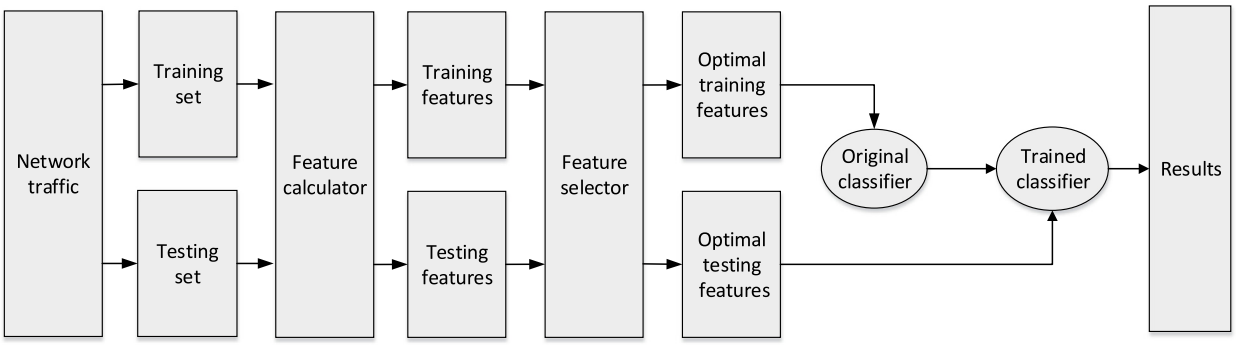

Fig. 4. Model of the method.

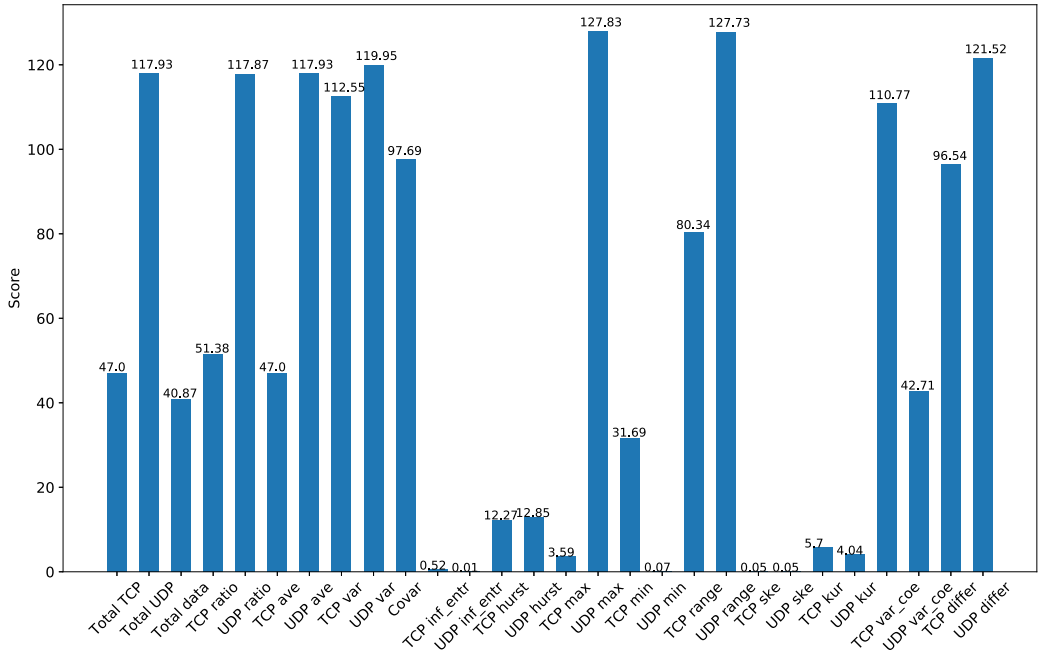

Fig. 5. Correlation scores between features and data labels.

and non-attack data. The original classifier obtains the ability to distinguish between LDoS attack flows and normal flows by learning and remembering the characteristics of the training set. The data of the testing set are used to evaluate the performance of the well-trained classifier. The training set and the testing set must be processed in two steps: feature calculation and feature selection. First, based on the feature set, the feature calculator converts the original traffic data of the training set and the testing set into feature data. In this process, the training feature data will be assigned labels. Each label represents the true category of the corresponding data. Furthermore, the feature data that are beneficial to the training of the classifier and can ensure the detection performance of the classifier will be retained by the feature selector. The optimal training features and training labels are used as input data of the original classifier to train the classifier. The optimal testing features are used as input data of the trained model to detect LDoS attacks.

4.2. Feature selection

Each network traffic feature in the feature set can reflect changes in the network. However, for network data that were acquired by different network environments, the data of each feature that were obtained by feature calculation differ in terms of their effects on the final detection results. Feature selection can screen out the most relevant features with detection results so that the detection algorithm can be effectively trained in the training stage and will yield more accurate detection results in the detection process [39].

4.2.1. Chi-squared test algorithm

In this paper, the Chi-squared test algorithm is used for feature selection [40,41]. The Chi-squared test is a hypothesis test method that is based on the χ^2 distribution, which is a classic data analysis algorithm in data mining. In machine learning, the Chi-squared test is often used for the correlation analysis of two categorical variables [42]. The proposed method uses the Chi-squared test algorithm to calculate the correlation between each feature and the data labels in the training stage. By comparing the scores of correlation, each feature will be assigned a priority, which will be the basis for feature selection.

The Chi-squared test algorithm can be expressed as,

$$\chi^2 = \sum \frac{(A-E)^2}{E} \tag{1}$$

Here, A represents an independent variable (eigenvalue), E represents a dependent variable (data label), and χ^2 is the deviation between the independent variable and the dependent

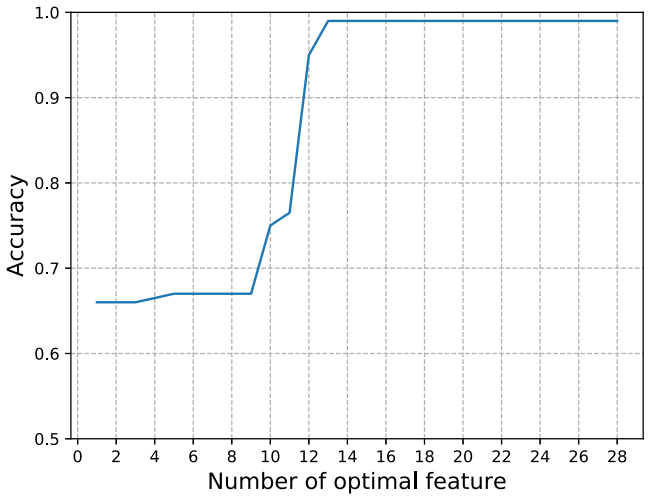

Fig. 6. Relationship between the number of optimal features and the detection accuracy.

variable. The smaller χ^2 is, the stronger the correlation between A and E is.

Fig. 5 plots the scores of correlation between each feature in the feature set and the data labels in an NS2 experimental scenario. The horizontal axis represents each feature in the feature set, and their positional order is consistent with the values in Table 1. The vertical axis represents the score of correlation, which is obtained by converting the P-value that is calculated by the Chi-squared test algorithm. The higher the score of the feature is, the stronger the correlation between the feature and the data label is. The results in Fig. 5 demonstrate that the influence on the detection results differs among features.

4.2.2. Number of optimal features

The priority of each feature is determined by the Chi-squared test algorithm. Features with higher priority are preferred as the optimal detection features. We determine the number of optimal features according to the detection accuracy of the classifier. In this process, the feature data with the highest priority are selected as the first input data of the classifier, and the corresponding detection result is obtained. Then, we gradually increase the number of features according to the priority level. Finally, the number of optimal features is selected by comparing the accuracies of detection. Fig. 6 plots the relationship between the number of optimal features and the detection accuracy in this process. The classifier is trained and obtains the detection results by cross-validation in the feature selection phase.

According to Fig. 6, with the increase in the number of optimal features, the accuracy of the classifier gradually increases. When the number of optimal features is 12, the classifier can realize a detection rate of more than 90%; hence, the classifier can recognize LDoS attacks. When the number of optimal features reaches 13, the classifier realizes the optimal detection rate. With the further increase of the number of feature, the detection rate of the classifier no longer improves; hence, the 13 optimal features yield the highest detection rate of the classifier. In addition, the features with lower priority have less of an impact on the detection results. Therefore, we abandon the data that correspond to these features with lower priority, which is regarded as noise data, to improve the training process of the classifier and to increase the detection rate in the detection process.

4.3. Improved Adaboost algorithm

Ensemble learning is an important branch of machine learning [43]. It performs a classification task by constructing and combining multiple weak classifiers to form a strong classifier. The core principles are that the classification ability of a weak classifier is limited and difficult to improve and that multiple weak classifiers can be combined to form a powerful and stable classifier. Two types of ensemble learning algorithms have been developed: in algorithms of the first type, weak classifiers have mutual dependencies, e.g., Boosting algorithm; in algorithms of the other type, weak classifiers are independent, e.g., Bagging algorithm and Random Forest algorithm.

4.3.1. Boosting

Boosting is a strategy in ensemble learning and includes the following main steps: First, a weak classifier is trained with the initial training set. Then, the distribution of the training samples is adjusted according to the results of the weak classifier so that the training samples that were misclassified by the previous weak classifier receive more attention from the next weak classifier. The previous steps are repeated until the number of weak classifiers reaches the set value T. Finally, T weak classifiers are weighted and combined to form a strong classifier. Fig. 7 presents a schematic diagram of the Boosting algorithm.

4.3.2. Adaboost

Adaboost is a typical Boosting algorithm. The Adaboost algorithm consists of two main parts: the addition model and the forward step-by-step algorithm. In the addition model, a strong classifier is a linear combination of a series of weak classifiers. The addition model can be expressed as follows:

$$H(x) = \sum_{t=1}^{T} \alpha_t h_t(x) \tag{2}$$

In this formula, $h_t(x)$ denotes a weak classifier, α_t is the weight of the weak classifier in the strong classifier, and $H(x)$ denotes a linear combination of weak classifiers.

In the forward step-by-step algorithm, the classifier that is generated by the next iteration is trained on the basis of the classifier from the previous iteration. It can be expressed as

$$H(x)_m = H(x)_{m-1} + \alpha_t h_t(x) \tag{3}$$

$h_t(x)$ represents the weak classifier in the tth iteration, α_t is the weight of the tth weak classifier, and $H(x)_{m-1}$ is the combination of all weak classifiers in the previous iteration.

The loss function that is used by the Adaboost algorithm is the exponential loss function. Therefore, the weight calculation formula for each round of weak classifiers is as follows:

$$\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{4}$$

Here, ϵ_t is the classification error rate of the tth iteration.

The distribution of the training samples is adjusted based on α_t. The formula is as follows:

$$w_{t+1} = \frac{w_t}{z_t} exp\left(-\alpha_t y h_t(x)\right) \tag{5}$$

where w_t represents the distribution weight of the previous round of training samples, y represents the classification label, and z_t is the normalization factor.

The final strong classifier is represented as

$$f(x) = sign\left(H(x)\right) \tag{6}$$

where *sign* is a symbolic function that converts the results that were obtained by the strong classifier into classification results.

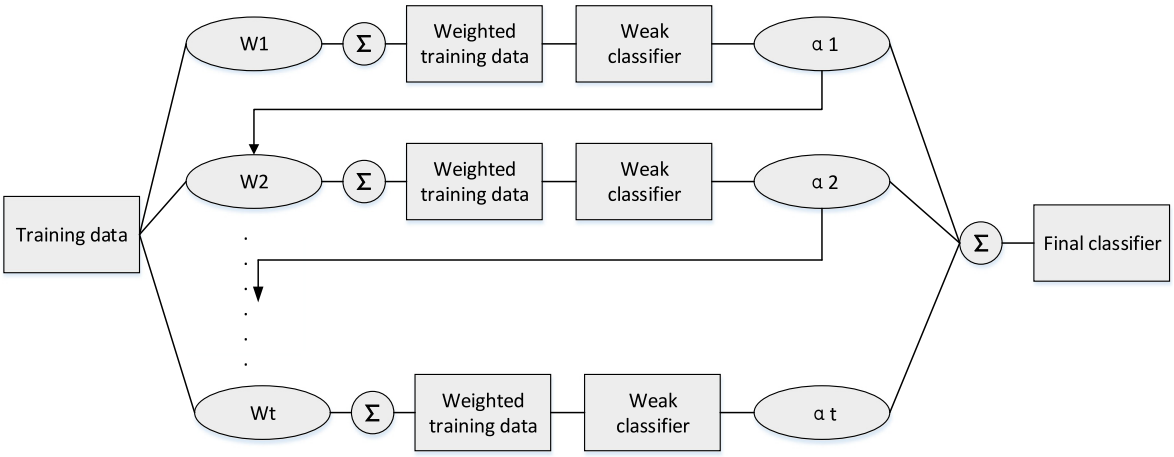

Fig. 7. Schematic diagram of the Boosting algorithm.

4.3.3. Improved Adaboost

In the process of training, the traditional Adaboost algorithm will increase the weights of the samples that are classified incorrectly in each iteration. These samples may be misclassified many times, which causes the weights of the samples to increase continuously and results in a serious imbalance of the weight distribution of the samples. To solve this problem, we improve the weight updating algorithm. We adjust the weight w_t that is obtained in the tth iteration to reduce the difference in the sample weights between the (t − 1)th iteration and the tth iteration to alleviate the imbalance of the sample weight distribution. The improved formula is as follows:

$$s_t = w_{t-1} + \frac{z \cdot e^{-|z|}}{1 + e^{-|z|}} \tag{7}$$

Here, $z = w_t - s_{t-1}$, w_{t-1} represents the weight that was obtained in the previous iteration, and s_t is the adjusted weight. The improved Adaboost algorithm is presented as Algorithm 1.

Algorithm 1 Pseudo code of the improved Adaboost algorithm.

Input: The sample data D; The number of weak classifier T;
Output: The strong classifier H;
1: Initializing sample weights w_1, $s_1 = w_1$;
2: **for** $t = 1$; $t < T$; $t++$ **do**
3: Training weak classifier h_t based on w_t and D;
4: Calculating the classification error rate ϵ_t;
5: Calculating the weight α_t of the weak classifier h_t, $\alpha_t = \frac{1}{2} ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right)$;
6: Updating weight w_{t+1} of D, $w_{t+1} = \frac{w_t}{z_t} exp\left(-\alpha_t y h_t\left(x\right)\right)$;
7: Adjusting weight w_{t+1}, $z = w_{t+1} - s_t$, $s_{t+1} = w_t + \frac{z \cdot e^{-|z|}}{1+e^{-|z|}}$, $w_{t+1} = s_{t+1}$;
8: Forming a new classifier $H_t = H_{t-1} + \alpha_t h_t$;
9: **end for**
10: $H = \sum_{t=1}^{T} \alpha_t h_t\left(x\right)$;
11: **return** H.

4.4. Detection algorithm description

First, we define the following concept:

Definition. Data slice (DS): a sample data sequence over a continuous period of time.

In the proposed method, the DS is the feature calculation unit and detection unit. Through the analyses of Sections 4.1, 4.2 and 4.3, the proposed method includes two parts: training of a detection model and attack detection. The training process of the detection model is presented in Algorithm 2 and the process of attack detection is presented in Algorithm 3.

Algorithm 2 Algorithm for the training of the detection model.

Input: The training set $D_{training\_set}$; The size of DS $time_{DS}$; The feature set $C_{feature\_set}$;
Output: The optimal feature set $C_{optimal}$; The well-trained classifier H;
1: Splitting $D_{training\_set}$ by $time_{DS}$ to get a data slice set $S_{training\_set}$ $(DS_1, DS_2, \cdots, DS_n)$;
2: Calculating traffic characteristics based on $C_{feature\_set}$ for each DS in $S_{training\_set}$ to obtain a feature data set $E_{training\_set}$ $(E_{DS_1}, E_{DS_2}, \cdots, E_{DS_n})$ and a label set $L_{training\_set}$ $(label_{DS_1}, label_{DS_2}, \cdots, label_{DS_n})$, E_{DS_n} represents the eigenvalue vector of DS_n, $label_{DS_n}$ represents the label of DS_n;
3: Selecting the optimal features with Chi-square test based on $E_{training\_set}$ and $L_{training\_set}$ to get an optimal feature set $C_{optimal}$ and an optimal feature data set $OE_{training\_set}$ $(OE_{DS_1}, OE_{DS_2}, \cdots, OE_{DS_n})$, OE_{DS_n} represents the optimal eigenvalue vector of DS_n;
4: Training H with $OE_{training\_set}$ and $L_{training\_set}$;
5: **return** H, $C_{optimal}$.

Algorithm 3 Algorithm for the attack detection.

Input: The duration of detection $time_{detection}$; The size of DS $time_{DS}$; The optimal feature set $C_{optimal}$; The well-trained classifier H;
Output: The detection results R;
1: Obtaining network traffic data with length $time_{detection}$ in real time to get a testing set $D_{testing\_set}$;
2: Splitting $D_{testing\_set}$ by $time_{DS}$ to get a data slice set $S_{testing\_set}$ $(DS_1, DS_2, \cdots, DS_n)$;
3: Calculating traffic characteristics based on $C_{optimal}$ for each DS in $S_{testing\_set}$ to obtain a feature data set $F_{testing\_set}$ $(F_{DS_1}, F_{DS_2}, \cdots, F_{DS_n})$, F_{DS_n} represents the eigenvalue vector of DS_n;
4: Detecting the $F_{testing\_set}$ with H to get the detection results R;
5: **return** R.

5. Experiments and evaluation

In this section, to evaluate the performance of the method that is proposed in this paper, we will conduct experiments on the NS2 simulation platform [44] and a test-bed platform.

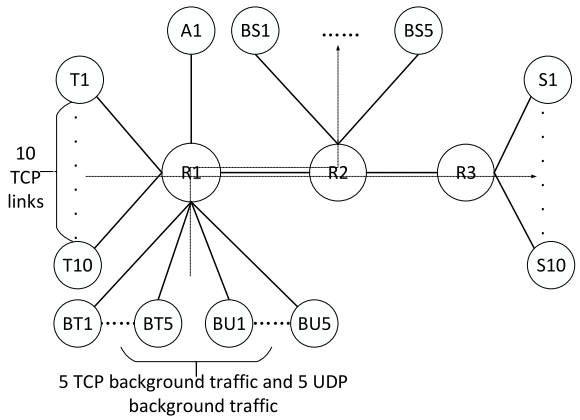

Fig. 8. Topology diagram of the NS2 simulation experimental platform.

Table 2
Bandwidth and network delay of each link on the NS2 experimental topology.

Node	Link bandwidth	Network delay
(T1–T10,R1)	100 Mbps	15 ms
(BT1–BT5,R1)	100 Mbps	15 ms
(BU1–BU5,R1)	100 Mbps	15 ms
(A1,R1)	100 Mbps	15 ms
(R1,R2)	100 Mbps	15 ms
(R2,BS1–BS5)	100 Mbps	15 ms
(R2,R3)	10 Mbps	30 ms
(R3,S1–S10)	100 Mbps	15 ms

5.1. Experiments on the NS2 simulation platform

We build a simulation network on the NS2 platform for conducting relevant experiments and obtaining the corresponding experimental data [45].

5.1.1. Experimental environment

The virtual network topology that we built on the NS2 platform is illustrated in Fig. 8. In this scenario, the backbone of the network consists of three routers: R1 (router 1), R2 (router 2) and R3 (router 3). R2 is the key router. The link between R2 and R3 is a bottleneck link. R1, R2 and R3 all adopt the RED queue management algorithm [46–48]. Nodes T1–T10 are used to send normal TCP traffic, and nodes S1–S10 are used to receive normal TCP traffic. Nodes T1–T10 and S1–S10 constitute ten normal TCP flows, which pass through R1, R2 and R3. Nodes BT1–BT5 are used to send TCP background traffic, and nodes BS1–BS5 are used to receive TCP background traffic. Nodes BT1–BT5 and BS1–BS5 form five TCP background traffic flows, which pass through R1 and R2. All TCP flows adopt the New Reno congestion control algorithm and set RTO to 1.0 s [49]. BU1–BU5 are used to send UDP background traffic. Background traffic is used to simulate the complexity of a real network. Node A1 represents the attacker, which periodically sends short-lived and high-pulsed UDP packets through R1, R2 and R3. The role of the data packet that is sent by A1 is to cause congestion on the bottleneck link, which results in packet loss in normal TCP flows. The bandwidths and network delays of all links are listed in Table 2.

5.1.2. Experimental parameters and data

On the NS2 simulation experimental platform, an LDoS attack is defined by a triple (T, t, R), where T represents the attack period, t represents the attack duration, and R represents the attack strength. We conducted three sets of experiments, each of which contained an LDoS attack with distinct attack parameters. The parameters of the LDoS attack are listed in Table 3. The

Table 3
LDoS attack parameters for each set of experiments.

Group ID	Attack cycle T (s)	Attack duration t (s)	Attack intensity R (Mb/s)	Cost of attack A
1	1	0.1	20	0.2
2	1	0.3	30	0.9
3	2	0.1	20	0.1

cost of attack measures the quality of the LDoS attack. A low cost of attack corresponds to superior concealment of the LDoS attack; however, the attack may not realize optimal effectiveness. Therefore, there is a trade-off between the cost of attack and the effectiveness of attack. As shown in Fig. 9, the second set of experiments has the strongest effectiveness of attack, but its concealment is the poorest. The LDoS attacks in the first and third sets of experiments can cause the network traffic to violently oscillate and to have a lower attack cost. The first group of experiments realized superior attack effects than the third group. The first group and the third group of experiments can realize the objective of the LDoS attack very well and the attack is not easy to detect. The calculation formula for the cost of attack is as follows:

$$A = \frac{t}{T} \times \frac{R}{C} \tag{8}$$

Here, A represents the cost of attack and C represents the bandwidth of the bottleneck link.

The experimental duration is 600 s in each set of experiments. The size of each data slice is 3 s. Therefore, each set of experiments will contain 200 data slices. The weak classifier that is used by the Adaboost algorithm is the Naive Bayesian classifier, and the number of weak classifiers is 100.

We evaluate the feasibility and the anti-interference performance of the proposed method with the data that were obtained from three groups of experiments.

5.1.3. Feasibility experiments on the proposed method

The flow data in each set of experiments are divided into two parts: The data from the first 300 s are used as a training set, and the data from the last 300 s are used as a testing set. Therefore, the training set and the testing set in each set of experiments each contain 100 data slices. In the testing set, the normal data slices are DS_1-DS_6, $DS_{21}-DS_{26}$, $DS_{41}-DS_{46}$, $DS_{61}-DS_{66}$ and $DS_{81}-DS_{100}$, and the attacked data slices are DS_7-DS_{20}, $DS_{27}-DS_{40}$, $DS_{47}-DS_{60}$ and $DS_{67}-DS_{80}$. For the labels of the data slices, we use 1 to represent attacked data slices and 0 to represent normal data slices.

According to Fig. 10, the detection algorithm can obtain the optimal detection result when the number of optimal features is 23 in the first set of experiments. Fig. 11 presents the detection results for the testing set in the first set of experiments. The red dotted line represents a data slice that is classified incorrectly. As shown in Fig. 11, the proposed method can realize a detection rate of 92.86%. DS_7, DS_{27}, DS_{47} and DS_{67} are incorrectly classified as normal data slices. DS_{21} and DS_{61} are incorrectly classified as attacked data slices.

Fig. 12 shows that the detection algorithm can realize the optimal detection result when the number of optimal features is 5 in the second set of experiments. Fig. 13 presents the detection results for the testing set in the second set of experiments. As shown in Fig. 13, the proposed method can realize a detection rate of 92.86%. DS_7, DS_{27}, DS_{47} and DS_{67} are incorrectly classified as normal data slices. DS_{21}, DS_{41}, DS_{61} and DS_{81} are incorrectly classified as attacked data slices.

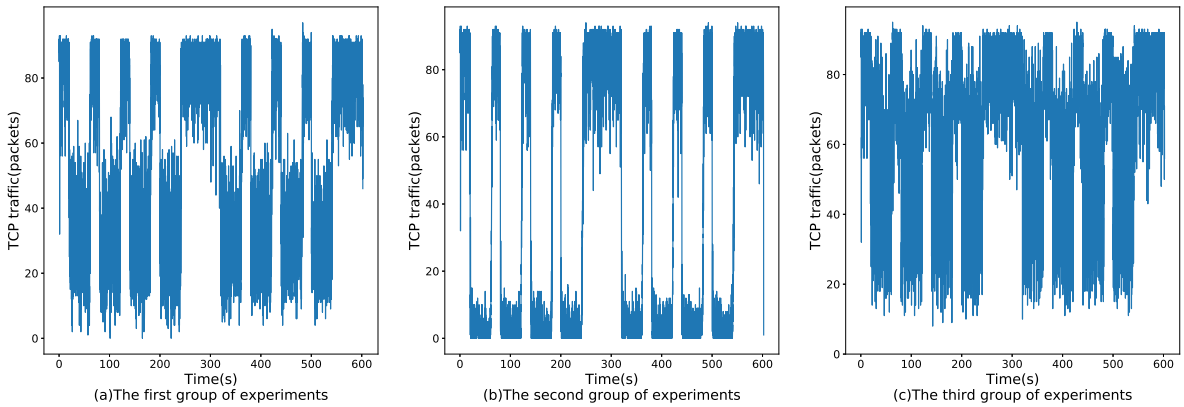

Fig. 9. TCP traffic on the NS2 experimental platform.

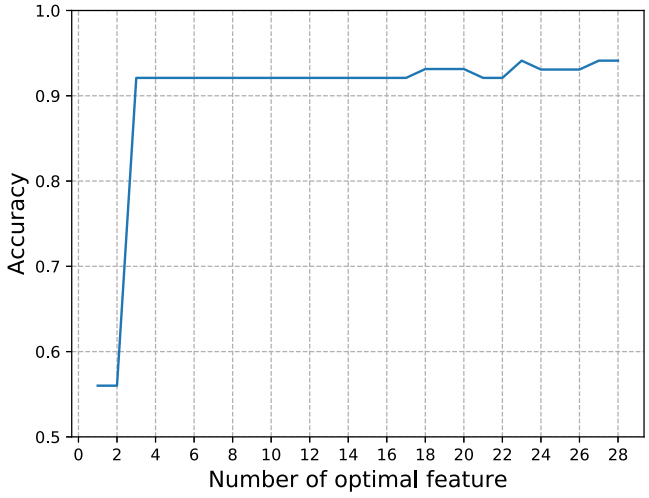

Fig. 10. Feature selection process for the first set of experiments on the NS2 platform.

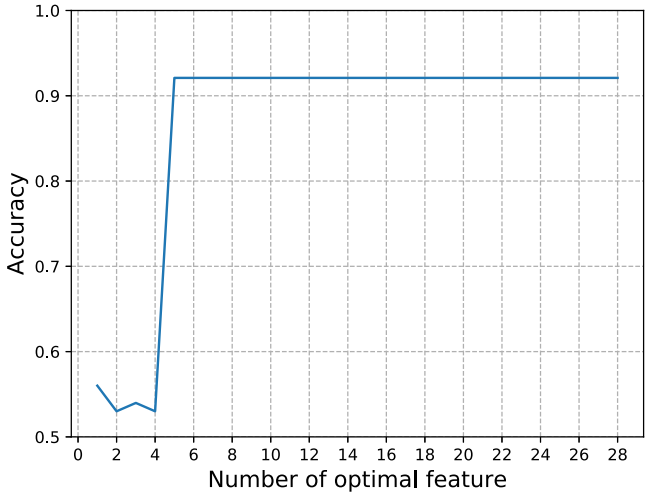

Fig. 12. Feature selection process for the second set of experiments on the NS2 platform.

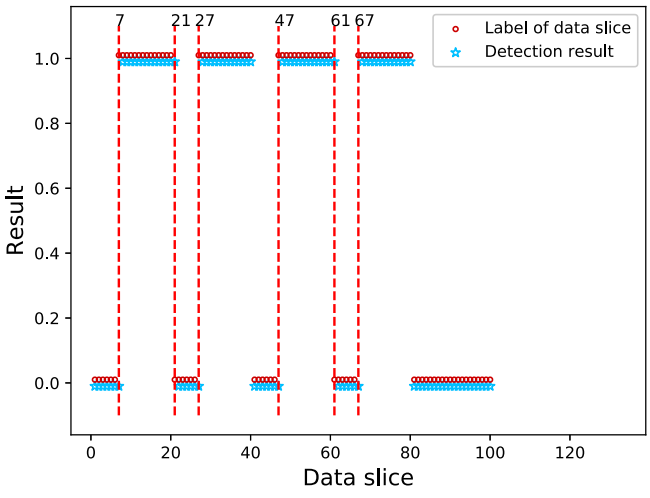

Fig. 11. Detection results of the first set of experiments on the NS2 platform.

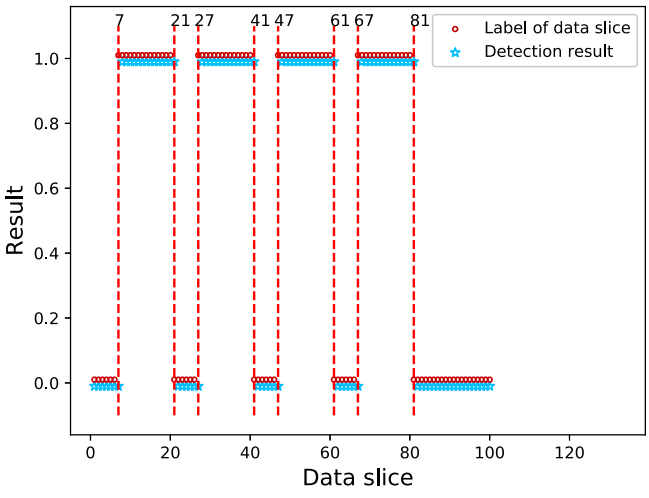

Fig. 13. Detection results of the second set of experiments on the NS2 platform.

According to Fig. 14, the detection algorithm can realize the optimal detection result when the number of optimal features is 27 in the third set of experiments. Fig. 15 presents the detection results for the testing set in the third set of experiments. As shown in Fig. 15, the proposed method can realize a detection rate of 96.43%. DS_7 and DS_{67} are incorrectly classified as normal data slices. DS_{21}, DS_{41}, DS_{61} and DS_{81} are incorrectly classified as attacked data slices.

Overall, our approach realized an average detection rate of 94.05% on the NS2 simulation platform. The experimental results demonstrate that our method can effectively detect LDoS

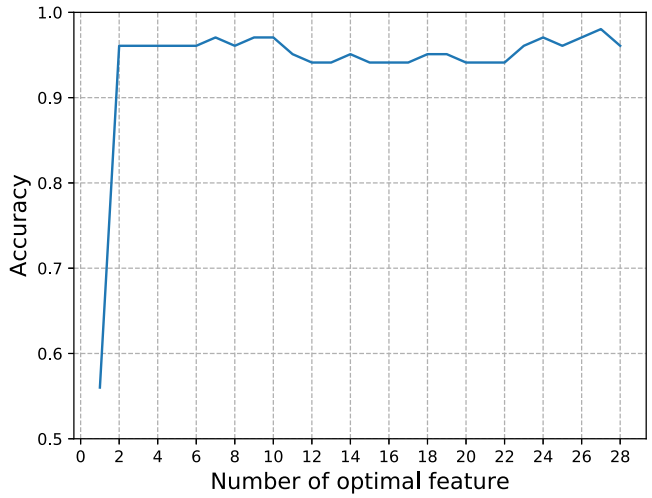

Fig. 14. Feature selection process for the third set of experiments on the NS2 platform.

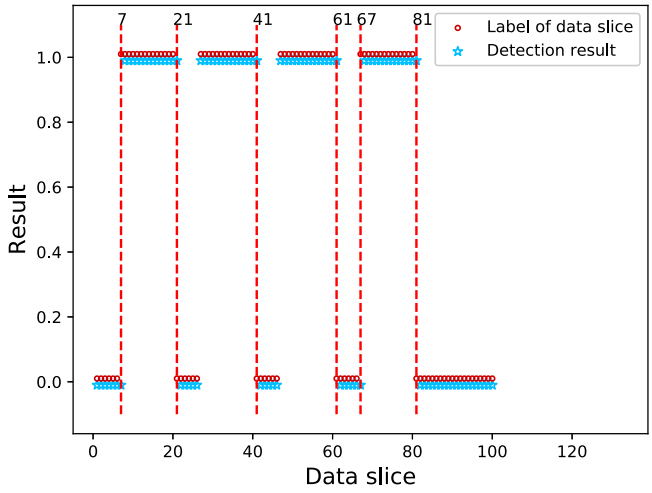

Fig. 15. Detection results of the third set of experiments on the NS2 platform.

attacks. The experiments evaluate the performance of the proposed method very effectively by setting up LDoS attacks with various attack strengths and attack costs. The data slices that were classified incorrectly in the experiments are located mainly at the boundary between normal traffic and abnormal traffic. This is because these data slices may contain some of the attack traffic but not all of it. Therefore, the eigenvalues that are calculated by these data slices will be between normal flow and abnormal flow, which causes the features to become blurred. Based on this, we can further detect the start time and the end time of the LDoS attack. This will be carried out in the future.

5.1.4. Anti-interference performance of the proposed method

We evaluated the anti-interference performance of the proposed method by using one set of data from the three sets of experiments as the training set and the other two sets of data as the testing set. The experimental results are presented in Table 4. In this experiment, our method realizes an average detection rate of 97.32%, a false-negative rate of 2.68% and a false-positive rate of 5.87%. The experimental results demonstrate that the proposed method has satisfactory anti-interference performance and the well-trained detection model can identify multiple LDoS attacks.

Table 4
Experimental results on the anti-interference performance of the proposed method.

Training set	Testing set	Detection rate	False-negative rate	False-positive rate
1	2,3	100%	0	7.95%
2	1,3	95.98%	4.02%	5.11%
3	1,2	95.98%	4.02%	4.55%

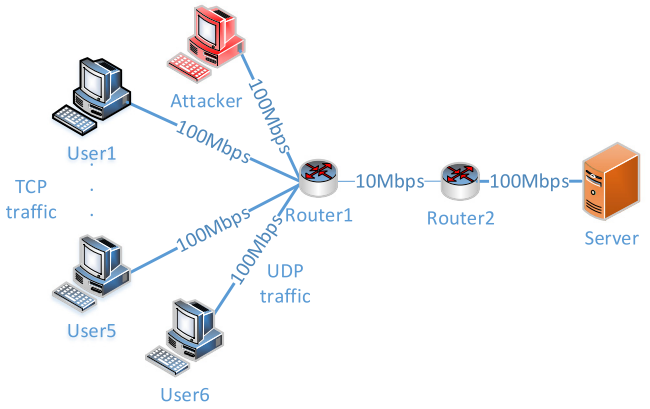

Fig. 16. Network topology diagram of the test-bed platform.

5.2. Experiments on a test-bed platform

We built a test-bed platform with real equipment for simulating LDoS attacks. The performance of our method is evaluated by acquiring data from the test-bed platform.

5.2.1. Experimental environment

The network topology of the test-bed platform that we built is illustrated in Fig. 16. In this topology diagram, there are two routers (Router1 and Router2), 6 clients (User1–User6), 1 Attacker and 1 Server. Router1 and Router2 are responsible for forwarding packets. Hosts User1–User5 send normal TCP packets to the Server. Host User6 sends normal UDP packets to the Server. Host Attacker performs an LDoS attack by periodically sending malicious UDP traffic to the Server. Host Server is used to receive packets from the clients and to send reply packet to the clients. Hosts User1–User6 and Attacker are connected to Router1, and host Server is connected to Router2. Router2 is a key router, which is the target of the LDoS attack. The link between Router 1 and Router 2 is a bottleneck link with a link bandwidth of 10 Mbps. Except for the bottleneck link, the bandwidth of all links is 100 Mbps. We use a socket program to establish the connections between the clients and Server.

5.2.2. Experimental parameters and data

On the test-bed platform, we define an LDoS attack by a triple (T, t, R), in which T represents the attack period, t represents the attack duration, and R represents the attack strength. In the socket program, we use multithread technology to send packages. Therefore, we adjust the strength of the LDoS attack by controlling the number of threads. The experimental parameters are listed in Table 5. We conducted three groups of experiments on the test-bed platform. Each group of experiment collects two sets of traffic data of the same duration. One set of data is used as the training set and the other set of data is used as the testing set. The size of each data slice is 3 s. For the labels of the data slices, we use 1 to represent attacked data slices and 0 to represent normal data slices. The weak classifier that is used by the Adaboost algorithm is the Naive Bayesian classifier, and the number of weak classifiers is 100.

Table 5
Experimental parameters of the test-bed platform.

Group ID	Cycle T (s)	Duration t (s)	Threads	Total time (s)	Attack time (s)
1	1	0.2	500	600	200–400
2	1	0.2	750	600	200–400
3	1	0.1	1000	900	300–600

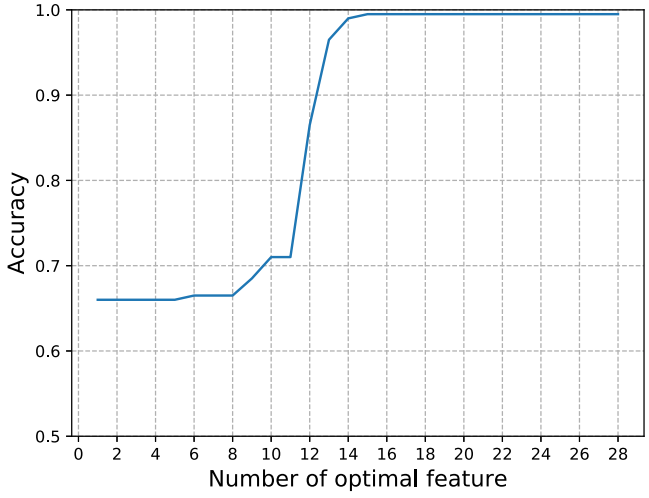

Fig. 17. Feature selection process for the first set of experiments on the test-bed platform.

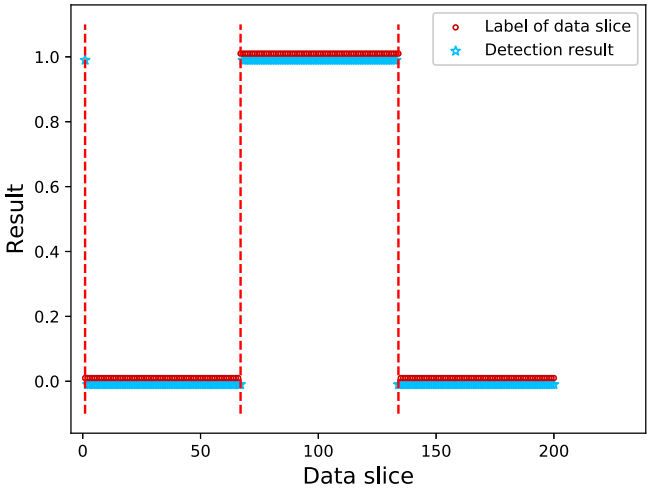

Fig. 18. Detection results of the first set of experiments on the test-bed platform.

5.2.3. Experimental results and analysis

According to Fig. 17, the detection algorithm can obtain the optimal detection result when the number of optimal features is 15 in the first set of experiments. Fig. 18 plots the detection results for the testing set in the first set of experiments. The red dotted line represents a data slice that is classified incorrectly. As shown in Fig. 18, the proposed method can realize a detection rate of 99%. The false-negative rate is 1% and the false-positive rate is 1%. DS_{67} and DS_{134} are incorrectly classified as normal data slices. DS_1 is incorrectly classified as an attacked data slice.

According to Fig. 19, the detection algorithm can obtain the optimal detection result when the number of optimal features is 15 in the second set of experiments. Fig. 20 plots the detection results for the testing set in the second set of experiments. As shown in Fig. 20, the proposed method can realize a detection

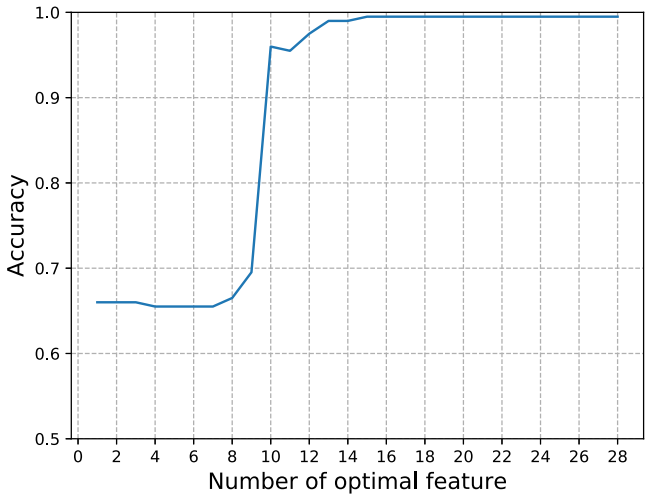

Fig. 19. Feature selection process for the second set of experiments on the test-bed platform.

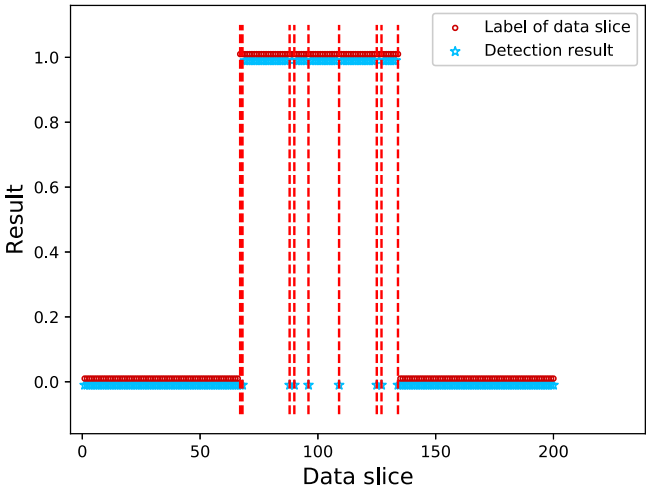

Fig. 20. Detection results of the second set of experiments on the test-bed platform.

rate of 95.5%, and the false-negative rate is 4.5%. DS_{67}, DS_{68}, DS_{88}, DS_{90}, DS_{96}, DS_{109}, DS_{125}, DS_{127} and DS_{134} are incorrectly classified as normal data slices.

According to Fig. 21, the detection algorithm can realize the optimal detection result when the number of optimal features is 15 in the third set of experiments. Fig. 22 plots the detection results for the testing set in the third set of experiments. As shown in Fig. 22, the proposed method can realize a detection rate of 96.67%, and the false-negative rate is 3.33%. DS_{158}, DS_{167}, DS_{177}, DS_{181}, DS_{184}, DS_{185}, DS_{187}, DS_{189}, DS_{194} and DS_{200} are incorrectly classified as normal data slices.

On the test-bed platform, the proposed method realized an average detection rate of 97.06%. The experimental results demonstrate that our method performs satisfactorily in detecting LDoS attacks on the test-bed platform. LDoS attacks in various scenarios can be effectively detected; hence, the performance of the proposed method is reliable and stable.

We further analyze the traffic of data slices that are classified incorrectly. As shown in Fig. 23, DS_{96} is an attacked data slice that is classified incorrectly, and DS_{100} is an attacked data slice that is classified correctly. In DS_{100}, the TCP traffic is in a volatile and unstable state, and the UDP traffic changes periodically. Therefore, DS_{100} is continuously attacked by the LDoS attack. In DS_{96},

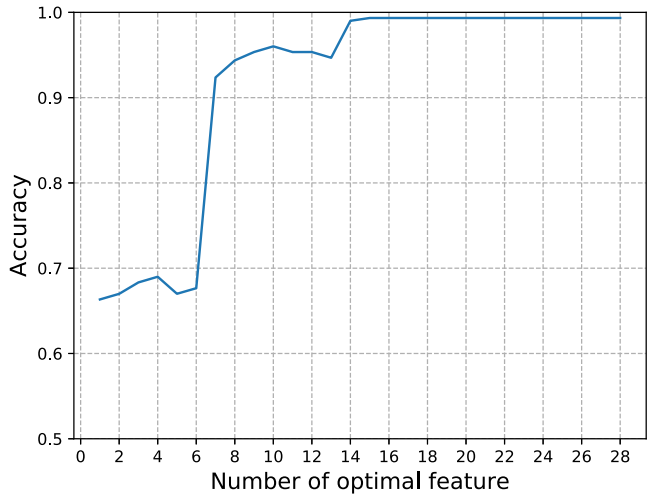

Fig. 21. Feature selection process for the third set of experiments on the test-bed platform.

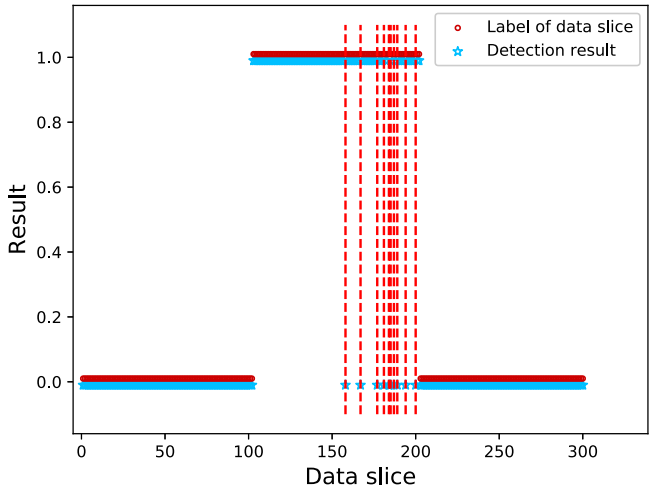

Fig. 22. Detection results of the third set of experiments on the test-bed platform.

Table 6
Comparison results.

Method	Detection rate	FNR	FPR
Multifractal	91%	9%	10%
BP neural network	96.68%	3.32%	3.89%
PSD	95.1%	4.9%	3.7%
MF-Adaboost	**97.06%**	**2.94%**	**0.33%**

there is a silent period of LDoS attack from 287 s to 288 s. During the silent period, both the TCP traffic and the UDP traffic are restored to a stable state. Therefore, DS_{96} contains 30∼40% of normal traffic, which causes the extracted traffic characteristics to be ambiguous.

5.3. Comparative experiments

We further compare our approach to other methods. For an objective and fair comparison, we select three methods that have been tested on the test-bed platform or on the real network for comparison with the proposed method. These methods are Multifractal [38], BP neural network [50] and PSD [19]. The comparison results are presented in Table 6.

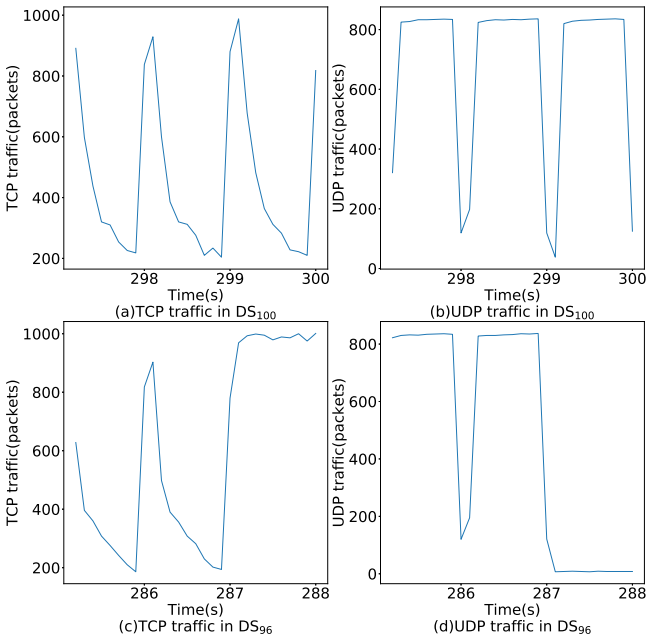

Fig. 23. Traffic in DS_{96} and DS_{100}.

According to Table 6, the proposed method outperforms the other three methods in terms of the detection rate and has a low false-negative rate and a low false-positive rate on the test-bed platform. Our approach has made satisfactory progress in the field of LDoS attack detection and can effectively detect LDoS attacks.

6. Conclusions and future work

In this paper, we propose an LDoS attack detection method that is based on MF-Adaboost. Based on an analysis of the network traffic, we construct a feature set of the network traffic. Based on the feature set, we process the original network traffic. The data processing of the network traffic includes feature calculation and feature selection. Feature calculation can extract abnormal changes in the network traffic and reduce the scale of the network data. The objective of feature selection is to select the features that are most conducive to the training of the classifier and to ensure that the classifier can obtain the optimal detection results. We use the Chi-squared test algorithm to select the features. In this paper, we use the Adaboost algorithm as the classifier of the detection model. The Adaboost algorithm is an adaptive and stable classification algorithm that can effectively deal with the complex and changeable network environment. Aiming at addressing the imbalance of the sample weights in the traditional Adaboost algorithm, we propose a sample weight adjustment algorithm. The sample weight adjustment algorithm can alleviate the imbalance of the sample weights, thereby avoiding the overfitting problem of the Adaboost algorithm in the training process. We conducted experiments on the NS2 simulation platform and the test-bed platform to evaluate the feasibility and performance of our method. On the NS2 simulation platform, the proposed method realized a detection rate of 94.05%. On the test-bed platform, the proposed method realized a detection rate of 97.06%. The experimental results demonstrate that our method can effectively detect LDoS attack traffic and has satisfactory stability.

In future work, we will evaluate the performance of our method on real networks. Based on the experimental results, we will determine the start time and end time of the LDoS attack. In addition, we will further improve the detection algorithm to realize the filtering of LDoS attack flows.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work was supported by National Natural Science Foundation of China (61772189 and 61702173), Hunan Provincial Natural Science Foundation of China (2019JJ40037 and 2018JJ3191), and the Scientific Research Fund of Hunan Provincial Education Department (18A178), and by FCT/MCTES through national funds and when applicable co-funded EU funds under the Project UIDB/EEA/50008/2020; and by Brazilian National Council for Scientific and Technological Development (CNPq) via Grant No. 309335/2017-5.

References

[1] K. Wen, J.H. Yang, B. Zhang, T. University, Survey on research and progress of low-rate denial of service attacks, J. Softw. 533 (7) (2014) 37.
[2] G. Somani, M.S. Gaur, D. Sanghi, M. Conti, Ddos attacks in cloud computing, Comput. Netw. Int. J. Comput. Telecommun. Netw. 109 (P2) (2016) 157–171.
[3] S. Yoon, H. Park, H.S. Yoo, Security issues on smarthome in iot environment, in: Computer Science and its Applications, Springer, 2015, pp. 691–696.
[4] Y. Yin, Y. Xu, W. Xu, G. Min, Y. Pei, Collaborative service selection via ensemble learning in mixed mobile network environments, Entropy 19 (7) (2017) 358.
[5] S. He, K. Xie, W. Chen, D. Zhang, J. Wen, Energy-aware routing for swipt in multi-hop energy-constrained wireless network, IEEE Access 6 (2018) 17996–18008.
[6] J. Wang, Y. Gao, X. Yin, F. Li, H.-J. Kim, An enhanced pegasis algorithm with mobile sink support for wireless sensor networks, Wirel. Commun. Mob. Comput. (2018) 1–9, http://dx.doi.org/10.1155/2018/9472075.
[7] J. Wang, C. Ju, Y. Gao, A.K. Sangaiahand, G.-j. Kim, A pso based energy efficient coverage control algorithm for wireless sensor networks, Comput. Mater. Continua 56 (3) (2018) 433–446.
[8] J. Wang, Y. Gao, W. Liu, A.K. Sangaiahand, H.-J. Kim, Energy efficient routing algorithm with mobile sink support for wireless sensor networks, Sensors 19 (7) (2019) 1494.
[9] T. Nguyen, J. Pan, T. Dao, An improved flower pollination algorithm for optimizing layouts of nodes in wireless sensor network, IEEE Access 7 (2019) 75985–75998.
[10] J. Wang, Y. Gao, A.K. Sangaiahand, H.-J. Kim, An improved routing schema with special clustering using PSO algorithm for heterogeneous wireless sensor network, Sensors 19 (3) (2019) 671–688.
[11] J. Wang, Y. Gao, W. Liu, W. Wu, S.-J. Lim, An asynchronous clustering and mobile data gathering schema based on timer mechanism in wireless sensor networks, Comput. Mater. Continua 58 (3) (2019) 711–725.
[12] S. Hameed, H. Ahmed Khan, Sdn based collaborative scheme for mitigation of ddos attacks, Future Internet 10 (3) (2018) 23.
[13] S. Nezhad, M. Nazari, E. Gharavol, A novel dos and ddos attacks detection algorithm using arima time series model and chaotic system in computer networks, IEEE Commun. Lett. 20 (4) (2016) 1.
[14] E. Bradley, H. Kantz, Nonlinear time-series analysis revisited, Chaos 25 (9) (2015) 097610.
[15] A. Sperotto, R. Sadre, A. Pras, Anomaly characterization in flow-based traffic time series, in: International Workshop on IP Operations and Management, Springer, 2008, pp. 15–27.
[16] Z.K. Gao, M. Small, J. Kurths, Complex network analysis of time series, Europhys. Lett. 116 (5) (2016) 50001.
[17] B. Krawczyk, L.L. Minku, M. Woniak, M. Woniak, M. Woniak, Ensemble learning for data stream analysis, Inf. Fusion 37 (C) (2017) 132–156.
[18] Z. Chen, C.K. Yeo, B.S. Lee, C.T. Lau, Power spectrum entropy based detection and mitigation of low-rate dos attacks, Comput. Netw. 136 (2018) 80–94.
[19] N. Agrawal, S. Tapaswi, Low rate cloud ddos attack defense method based on power spectral density analysis, Inform. Process. Lett. 138 (2018) 44–50.
[20] Zhijun, Minxiao, Wang, Changcan, Meng, Low-rate dos attack flows filtering based on frequency spectral analysis, China Commun. 14 (6) (2017) 98–112.
[21] X. Wu, D. Tang, L. Tang, J. Man, S. Zhan, Q. Liu, A low-rate dos attack detection method based on Hilbert spectrum and correlation, in: 2018 IEEE SmartWorld, Ubiquitous Intelligence &Amp; Computing, Advanced &Amp; Trusted Computing, Scalable Computing &Amp; Communications, Cloud &Amp; Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), IEEE, 2018, pp. 1358–1363.
[22] Y. Meng, L. Liang, Z. Wu, M. Wang, Identifying ldos attack traffic based on wavelet energy spectrum and combined neural network, Int. J. Commun. Syst. 31 (4) (2017) e3449.
[23] G. Yi, J. Yan, Q. Han, L. Zhang, A crf-theory-based method for bgp-ldos attack detection, in: IEEE International Conference on Computer & Communications, 2016.
[24] K. Xie, C. Peng, X. Wang, G. Xie, J. Wen, J. Cao, D. Zhang, Z. Qin, Accurate recovery of internet traffic data under variable rate measurements, IEEE/ACM Trans. Netw. 26 (3) (2018) 1137–1150.
[25] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, J. Cao, D. Zhang, Fast tensor factorization for accurate internet anomaly detection, IEEE/ACM Trans. Netw. 25 (6) (2017) 3794–3807.
[26] K. Xie, L. Wang, X. Wang, G. Xie, J. Wen, G. Zhang, J. Cao, D. Zhang, K. Xie, X. Wang, et al., Accurate recovery of internet traffic data: A sequential tensor completion approach, IEEE/ACM Trans. Netw. 26 (2) (2018) 793–806.
[27] K. Xie, X. Li, X. Wang, J. Cao, G. Xie, J. Wen, D. Zhang, Z. Qin, Online anomaly detection with high accuracy, IEEE/ACM Trans. Netw. (ISSN: 1063-6692) 26 (3) (2018) 1222–1235.
[28] S. He, Z. Li, Y. Tang, Z. Liao, J. Wang, H.-J. Kim, Parameters compressing in deep learning, Comput. Mater. Continua (2019).
[29] J. Zhang, X. Jin, J. Sun, J. Wang, A.K. Sangaih, Spatial and semantic convolutional features for robust visual object tracking, Multimedia Tools Appl. (2018) http://dx.doi.org/10.1007/s11042-018-6562-8.
[30] J. Zhang, C. Lu, J. Wang, X.-G. Wang, Concrete cracks detection based on fcn with dilated convolution, Appl. Sci. 9 (13) (2019) 2686.
[31] J. Zhang, C. Lu, X. Li, H.-J. Kim, J. Wang, A full convolutional network based on densenet for remote sensing scene classification, Math. Biosci. Eng. 16 (5) (2019) 3345–3367.
[32] J. Zhang, W. Wang, C. Lu, J. Wang, A.K. Sangaih, Lightweight deep network for traffic sign classification, Ann. Telecommun. (2019) http://dx.doi.org/10.1007/s12243-019-00731-9.
[33] J. Zhang, X. Jin, J. Sum, J. Wang, K. Li, Dual model learning combined with multiple feature selection for accurate visual tracking, IEEE Access 7 (1) (2019) 43956–43969.
[34] Z. Wu, Q. Pan, M. Yue, L. Liu, Sequence alignment detection of tcp-targeted synchronous low-rate dos attacks, Comput. Netw. 152 (2019) 64–77.
[35] M.H. Bhuyan, E. Elmroth, Multi-scale low-rate ddos attack detection using the generalized total variation metric, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 1040–1047.
[36] D. Tang, R. Dai, L. Tang, S. Zhan, J. Man, Low-rate dos attack detection based on two-step cluster analysis, in: International Conference on Information and Communications Security, Springer, 2018, pp. 92–104.
[37] Z. Wang, W. Yan, T. Oates, Time series classification from scratch with deep neural networks: A strong baseline, in: 2017 International Joint Conference on Neural Networks (IJCNN), IEEE, 2017, pp. 1578–1585.
[38] Z. Wu, L. Zhang, M. Yue, Low-rate dos attacks detection based on network multifractal, IEEE Trans. Dependable Secure Comput. 13 (5) (2016) 559–567.
[39] J. Cai, J. Luo, S. Wang, S. Yang, Feature selection in machine learning: A new perspective, Neurocomputing 300 (2018) 70–79.
[40] D. Shi, C. Distefano, H.L. Mcdaniel, Z. Jiang, Examining chi-square test statistics under conditions of large model size and ordinal data, Struct. Equation Model. Multidiscip. J. (3) (2018) 1–22.
[41] I.S. Thaseen, C.A. Kumar, Intrusion detection model using fusion of chi-square feature selection and multi class SVM, J. King Saud Univ.-Comput. Inf. Sci. 29 (4) (2017) 462–472.
[42] A. Peterkova, M. Nemeth, G. Michalconok, A. Bohm, Computing importance value of medical data parameters in classification tasks and its evaluation using machine learning methods, in: Computer Science on-Line Conference, 2018.
[43] T.G. Dietterich, Ensemble methods in machine learning, in: Proc International Workshgp on Multiple Classifier Systems, Vol. 1857 (1)(2000) pp. 1–15.
[44] S. He, K. Xie, K. Xie, C. Xu, W. Jin, Interference-aware multi-source transmission in multi-radio and multi-channel wireless network, IEEE Syst. J. (2019) http://dx.doi.org/10.1109/JSYST.2019.2910409.
[45] F.T. AL-Dhief, N. Sabri, N. Latiff, N. Malik, M. Abbas, A. Albader, M.A. Mohammed, R.N. AL-Haddad, Y.D. Salman, M. Khanapi, et al., Performance comparison between tcp and udp protocols in different simulation scenarios, Int. J. Eng. Technol. 7 (4.36) (2018) 172–176.
[46] M. Mehta, G. Deep, M. Mehta, Comparison of active queue management in ns2, Asian J. Comput. Sci. Inf. Technol. 7 (2017) 79–84.

[47] D.A. Alwahab, S. Laki, A simulation-based survey of active queue management algorithms, in: Proceedings of the 6th International Conference on Communications and Broadband Networking, ACM, 2018, pp. 71–77.
[48] M.U. Rahman, Z.U. Rahman, M. Fayaz, S. Abbas, R.K. Shahsani, Performance analysis of tcp/aqm under low-rate denial-of-service attacks, in: International Conference on Inventive Computation Technologies, 2017, pp. 1–5.
[49] A. Sembiring, M. Abdurohman, F. Yulianto, Tcp lr-newreno congestion control for ieee 802.15. 4-based network, Int. J. Intell. Eng. Syst. 10 (5) (2017) 181–190.
[50] Z.-j. Wu, J.-a. Zhang, M. Yue, C.-f. Zhang, Approach of detecting low-rate dos attack based on combined features, J. Commun. 38 (5) (2017) 19–30.

Dan Tang is a lecturer of College of Computer Science and Electronic Engineering (CSEE) Hunan University (HNU), Changsha, China. He received the Ph.D. degree in 2014. His research interests include the areas of computer network security, computer information security, and architecture of future Internet.

Liu Tang received the BA degree in network engineering from Hunan University of Science and Technology, China, in June 2017. He is currently a postgraduate in College of Computer Science and Electronic Engineering (CSEE) Hunan University (HNU), Changsha, China. His current research interests are cyber-space Security.

Rui Dai received the BA degree in computer science and technology from Hunan normal university, China, in June 2018. He is a currently postgraduate in College of Computer Science and Electronic Engineering (CSEE) Hunan University (HNU), Changsha, China. His current research interests are network attack detection.

Jingwen Chen entered Hunan University in China in September 2015. She is currently a senior in College of Computer Science and Electronic Engineering (CSEE) Hunan University (HNU), Changsha, China. Her research direction is network information security.

Xiong Li received the Ph.D. degree in computer science and technology from the Beijing University of Posts and Telecommunications, Beijing, China, in 2012. He is currently a Professor with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored over 100 referred papers. His current research interests include cryptography and information security. He was a recipient of the 2015 Journal of Network and Computer Applications Best Research Paper Award.

Joel J. P. C. Rodrigues is a professor at the Federal University of Piauí, Brazil; and senior researcher at the Instituto de Telecomunicações, Portugal. Prof. Joel is the leader of the Next Generation Networks and Applications Research Group (UFPI), Director for Conference Development - IEEE ComSoc Board of Governors, and IEEE distinguished Lecturer. He has authored or coauthored over 800 papers in refereed international journals and conferences, 3 books, 2 patents, and 1 ITU-T Recommendation. He had been awarded several Outstanding Leadership and Outstanding Service Awards by IEEE Communications Society and several best papers awards. Prof. Rodrigues is a member of the Internet Society, a senior member ACM, and Fellow of IEEE.