Computers & Security 88 (2020) 101645

Contents lists available at ScienceDirect

Computers & Security

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

A dynamic MLP-based DDoS attack detection method using feature selection and feedback

Meng Wang*, Yiqin Lu, Jiancheng Qin

School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China

ARTICLE INFO

Article history:
Received 24 June 2019
Revised 21 September 2019
Accepted 10 October 2019
Available online 10 October 2019

Keywords:
DDoS attacks
Feature selection
Intrusion detection
Multilayer perceptrons
Network security

ABSTRACT

Distributed Denial of Service (DDoS) attack is a stubborn network security problem. Various machine learning-based methods have been proposed to detect such attacks. According to our survey, the features used to characterize the attack are usually selected manually according to some personal understanding, and the detection model is expected to perform good generalization performance in practical detection all the time. Therefore, how to select the optimal features that perform the best performance is a critical problem for constructing an effective detector. Meanwhile, as network traffic gets increasingly complex and changeable, some original features may become incapable of characterizing current traffic, and detector failure could occur when traffic changes. In this paper, we chose the multilayer perceptrons (MLP) to demonstrate and solve the proposed problem. In our solution, we combined sequential feature selection with MLP to select the optimal features during the training phase and designed a feedback mechanism to reconstruct the detector when perceiving considerable detection errors dynamically. Finally, we validated the effectiveness of our method and compared it with some related works. The results showed that our method could yield comparable detection performance and correct the detector when it performed poorly.

1. Introduction

DDoS attacks have become one of the most stubborn network security problems for many years since the Computer Incident Advisory Capability reported the first attack incident in 1999 (Criscuolo, 2000). Although various defense methods have been proposed in academia and industry, the threat of DDoS attacks is still very severe and grows year by year. According to the report of Arbor Networks Inc. (2018), DDoS attacks still represented the dominant threat observed by the vast majority of service providers, and the largest attack in 2017 was 600 Gbps. DDoS attacks aim to stop legitimate users from normally accessing a specific network service through simultaneously and continuously sending a large amount of traffic to the target system (Manavi, 2018). To achieve this attempt, hackers usually use Botnets to launch a DDoS attack. Botnets are the network formed by "enslaving" host computers, called bots, that are controlled by one or more attackers, called botmasters, with the intention of performing malicious activities (Silva, 2013). Recently, the most powerful botnets tend to be based on Internet of Things (IoT) devices, as billions of vulnerable IoT devices have been deployed and connected and most IoT devices are easy to hack and compromise (Khan and Salah, 2018).

In the DDoS attack detection research community, detection methods are proposed based on different models and theories. Machine learning, information theory, and statistical models are the three leading methods that form the basis of the majority of present-day detection techniques (Singh et al., 2017). Machine learning techniques in cybersecurity are helpful by recommending the proper decision for analysis and even doing the proper action automatically, such as artificial neural networks (ANN), Bayesian network, decision trees (DT), support vector machine (SVM), clustering, ensemble learning and so on Hosseini and Azizi (2019). Various ANN models have been used in the area of the intrusion detection system (IDS), and they have many advantages in detecting the DDoS attack, including self-learning, self-organizing, better fault tolerance, robustness, and parallelism. In this paper, we focus on the ANN models and choose the MLP as a model to demonstrate and solve the proposed problem. We start by implementing an MLP-based detector and test it with open traffic data. The results show that the original features used in the detector have some redundancy, and removing a specific feature can improve the

* Corresponding author.
E-mail address: w.m15@mail.scut.edu.cn (M. Wang).

https://doi.org/10.1016/j.cose.2019.101645

detection accuracy. We then continue to survey more related works that used six different ANN models and find that the features used in the models are usually designed manually according to some personal and intuitive understanding of DDoS attacks rather than representation learning. Due to the different personal understanding of DDoS attacks, the same features may be useful in some cases but invalid in other cases. There is no universal criterion to follow about how to design the feature. Moreover, giving some importance to the irrelevant features will lead to a poor generalization on the new data, even if we try to control the overfitting in the training phase (Romero and Sopena, 2008). Therefore, we think feature selection is the most critical consideration for the machine learning-based DDoS attack detection. We need to select the optimal features through experimental approaches. For this purpose, we survey more related works about combining feature selection with ANN and other machine learning models. We aim to find an appropriate strategy that is feasible and can directly interact with the detection model to cater to our requirements.

On the other hand, according to our experience, a conventional way to evaluate a detection method is usually through providing an excellent detection accuracy on labeled test data. The potential hypothesis is that the tuned detector can always perform good generalization and correctly detect the following attacks in real detection. This way is reasonable when we use machine learning techniques to solve the pattern recognition problem, and we usually treat the DDoS attack detection as a binary or multiclass classification problem. However, network traffic becomes more and more complex and changeable, and DDoS attacks are also evolving day by day. The hypothesis is actually challenging to satisfy. In other words, some new normal patterns or some new unknown attacks may deviate from the patterns learned from the original train data, and the detector could make a lot of errors (including false-positive and false-negative errors) during real detection. In order to solve this problem, we need to design a method to perceive the detection errors timely and reconstruct the detector dynamically.

The rest of this paper is organized as follows: In Section 2, we review the related works. Section 3 describes the research problem. Section 4 describes the proposed solution. At last, we describe the experiment in Section 5 and our conclusion in Section 6.

2. Related works

2.1. ANN-based detection

We surveyed some proposed DDoS attack detection methods based on six different ANN models, including MLP, probabilistic neural network (PNN), radial basis function neural network (RBFNN), learning vector quantization (LVQ), recurrent neural networks (RNN), and self-organizing map (SOM). The former five models are supervised, and the last one is unsupervised al. A simple summary is listed in Table 1.

From Table 1, we could see that all works have excellent detection accuracies (the best could achieve about 99.11%, and the worst is not lower than 93.4%), but the features used in each ANN model are very different in the aspects of dimension and definition. The major innovations of these ANN-based methods are reflected in combing newly defined features with different models and deploying in different application scenarios. The authors of these papers only evaluate the detection performance on a given test data but ignore validating the necessity of the features through some experimental approaches. Besides, the availability of a detector is also limited by the features as they are extracted in different manners (see the feature description in Table 1). In summary, features are the foundation of an effective detection model in the machine learning-based detection research community, and detection performance totally depends on how well the features characterize the attacks. Therefore, how to select the optimal features is a critical problem for constructing a suitable detector. Feature selection will not only enhance the generalization performance and reduce the computing complexity, but also improve the availability of a detection model when it is deployed in different network environments.

2.2. Feature selection

Feature selection is one of the key problems for machine learning and data mining (Li et al., 2017b). The objective of feature selection is to select a feature subset that performs the best under a certain evaluation criterion (Romero and Sopena, 2008). According to the review of paper Li et al. (2017b), Li et al. (2017), Bolón-Canedo et al. (2013), feature selection can be categorized as wrapper, filter, and embedded methods from the strategy perspective. A typical wrapper method tries to find an optimal subset through iteratively evaluating the model with different combinations of features as inputs. The wrapper method interacts with the models, but the filter method, which assesses the feature saliency according to the intrinsic characteristics of data, is independent of the models. The embedded method is a tradeoff between the filter method and the wrapper method. As we focus on MLP and need to interact with it, a wrapper method is more suitable to combine with the detection model.

First, we introduce some works about feature selection for ANN models. Paper Windeatt et al. (2011) proposed a feature selection method using the evaluation criteria based on weight values. Paper Vesa et al. (2001), Monirul Kabir et al. (2010) proposed two wrapper-based feature selection methods. Paper Yusof et al. (2018), Osanaiye et al. (2016) used sequential backward selection (SBS), which means incapable features are excluded from the full feature set sequentially. Paper Wang et al. (2017) used sequential forward selection (SFS), which is contrary to the SBS. Paper Baesens et al. (2000) proposed a clamping technique to evaluate the relevance of a feature by fixing the input value of this feature to its mean. Paper Tang et al. (2014) added a preselecting module before the input layer based on mutual information, which is a filter method, and this paper also used a random strategy (i.e., genetic algorithm) to search the optimal feature subset. Paper Ji et al. (2016) used a variance analysis approach to find the most important features for the MLP model. Paper De La Hoz et al. (2014) proposed a feature selection method based on multi-objective optimization for the SOM model, and paper De la Hoz et al. (2015) assembled principal component analysis (PCA) and Fisher discriminant ratio (FDR) for a probabilistic SOM model to select the optimal features. Due to the lack of more recent literature, although some referenced works are outdated, they still provide some experience about combing feature selection with ANN models. According to our analysis, the weight-based selection method is unreliable and obscure, and the random strategy is computationally costly. As the complexity of searching the space with n features is $O(2^n)$ huge, finding an optimal feature subset through brute force search is infeasible when n is too large, but the sequential selection strategy reduces the complexity to $O(n^2)$. Hence, it is easily feasible to interact with the detection model.

Second, we extend our survey to some works about feature selection for machine learning-based IDS, which include detecting the Botnets and DoS/DDoS attacks. The details of the related papers are listed in Table 2. From Table 2, we know that filter, wrapper, and embedded feature selection methods are widely used to combine with machine learning models for intrusion detection. The difference among these works mainly comes from features and the combinations between feature selection methods and machine learning algorithms. Paper Beigi et al. (2014) evaluated the effectiveness of 21 different features that were used in

Table 1
Summary of ANN-based detection.

Model	Work	NoF	BestAcc	Feature Description
MLP	Saied et al. (2016)	5	98%	source IP address, TCP sequence, source port, destination port, TCP flags
	Chang-Jung and Ting-Yuan (2016)	7	94%	number of packets, average packet size, time interval variance, number of bytes
	Zhao et al. (2015)	3	-	CPU usage, average packet size, number of TCP connection
	Perakovic et al. (2017)	5	95.6%	packet arrival time, protocol, packet length, source IP address, destination IP address
	Saad et al. (2016)	5	98.3%	capture time, source IP address, destination IP address, packet length, protocol
	Singh et al. (2016)	3	98.31%	the number of HTTP GET, entropy of HTTP GET, variance of HTTP GET
	Cui et al. (2016)	5	-	number of packets matched by each flow entry, number of bytes matched by each flow entry, survival time of each flow entry, packet rate of each flow entry, bytes rate of each flow entry
	Chen and Yu (2016)	6	94.6%	total packet number, the proportion of ICMP packets in total packets, the proportion of short packets in total packets, the proportion of long packets in total packets, the proportion of UDP packets in total packets,$\log_{10}(SYNpacketnumber/ACKpacketnumber)$
	Ji et al. (2016)	22	96.67%	22 features of the 41 NSL-KDD features
	Kim and Gofman (2018)	41	98.5%	41 NSL-KDD features
	Kushwah and Ali (2018)	41	96.3%	41 NSL-KDD features
PNN	Akilandeswari and Shalinie (2012)	7	97%	source IP address, prefix of source IP, flow duration, flow number per time interval, packet size, packet number per flow, unique IP address per time interval
LVQ	Li et al. (2010)	8	99.7%	CPU usage, memory usage, system time, number of TCP connections, sent bytes, received bytes, number of passive TCP connections
RBF	Chen et al. (2012)	3	-	average time of IP showing time, dispersion of source IP, dispersion of source MAC
RNN	Chuan-long et al. (2017)	41	93.4%	41 NSL-KDD features
SOM	Braga et al. (2010)	6	99.11%	average of packets per flow, average of bytes per flow, average of duration per flow, percentage of pair-flows, growth of single-flows in time interval, growth of different ports in time interval
	Stevanovic et al. (2013)	10	-	click number, HTML-to-image ratio, percentage of PDF/PS file requests, percentage of 4xx error responses, percentage of HTTP requests in type HEAD, percentage of requests in unassigned referrers, number of bytes requested from the server, page popularity index, standard deviation of requested page's depth, percentage of consecutive sequential HTTP requests
	De La Hoz et al. (2014)	41	99.12%	41 NSL-KDD features
	De la Hoz et al. (2015)	41	94%	41 KDD'99 features

many previous Botnet detection works according to the results of a DT-based detection. Paper Osanaiye et al. (2016) assembled four filter selection methods to vote the optimal features. Paper Barati et al. (2014), Sindhu et al. (2012) combined a genetic algorithm with two different classifiers to randomly search the optimal features. Paper Wang et al. (2017), Lin et al. (2012), Li et al. (2012), Bolón-Canedo et al. (2011), Ji et al. (2016), De la Hoz et al. (2015), Kuang et al. (2014) combined several different filter or wrapper methods with SVM respectively. In summary, combing feature selection with a detection model is popular in the IDS research community, and this is very useful to improve the detection performance. Besides, one main research trend in the community is exploring the various hybrid methods to get better detection performance in a particular scenario.

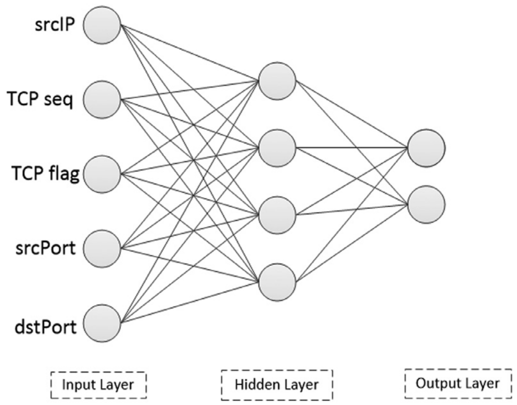


Fig. 1. TCP topological structure.

3. Problem description

In this section, we describe the research problem by implementing an MLP-based detection method proposed in paper Saied et al. (2016). This paper used three MLP models with different topological structures to process TCP, UDP, and ICMP packets. For brevity, we only implemented the TCP topological structure (see Fig. 1). The input layer of the model has five features, including source IP address, TCP sequence, TCP flag, and TCP source and destination port. We used an open traffic data named ISOT (Saad et al., 2011) to train and test the model. The training and testing process was operated on the Tensorflow platform. The experiment results on test data are shown in Tables 3 and 4. In the tables, the "topology" column denotes the topological structure of the MLP model. For example, "5-4-4-2" means the model has one input layer with five neurons, two hidden layers with four and four neurons respectively, and one output layer with two neurons. In this paper, we evaluate the detection performance through four metrics: accuracy (Acc), precision (Pre), detection rate (DR, also called recall), and false alarm rate (FAR) (Li et al., 2017a). These metrics are computed based on a confusion matrix (see Table 5), and they are defined as

$$Acc = \frac{TP+TN}{TP+FN+TN+FP}, \tag{1}$$

$$Pre = \frac{TP}{TP+FP}, \tag{2}$$

$$DR = \frac{TP}{TP+FN}, \tag{3}$$

$$FAR = \frac{FP}{FP+TN}. \tag{4}$$

From Table 3, we could see that changing the MLP structure does not bring significant improvement to the performance. These results indicate that the model structure slightly influences the detection performance. In Table 4, we sequentially removed ev-

Table 2
Summary of feature selection for ML-based IDS.

Work	Object	Feature Selection	Detection Model	BestAcc	Traffic Data
Beigi et al. (2014)	Botnet	Exclusion and Inclusion	DT: C4.5	99%	ISOT, ISCX, MCFP
Yusof et al. (2018)	DoS	Consistency-based, Characteristic-based	MLP	91.7%	NSL-KDD
Hosseini and Azizi (2019)	DDoS	forward feature selection	hybrid supervised learning algorithm	98.9%	NSL-KDD
Osanaiye et al. (2016)	DDoS	Information Gain, Gain Ration, Chi-squared, RelifF (Voting)	DT: J48	99.67%	NSL-KDD
Wang et al. (2017)	DDoS	Random Forest, ID3	SVM	82.99%	KDD'99
Balkanli et al. (2015)	DDoS	Chi-squared, Symmetrical Uncertainty	DT: C4.5	95%	CAIDA
Suresh and Anitha (2011)	DDoS	Chi-squared, Information Gain	Naive Bayes(NB), DT: C4.5, K-means, SVM, k-NN, Fuzzy c-means	98.7%	CAIDA
Barati et al. (2014)	DDoS	Genetic Algorithm	MLP	99.99%	CAIDA
Lin et al. (2012)	IDS	Simulated Annealing	SVM, DT	99.96%	KDD'99
Li et al. (2012)	IDS	SBS	SVM	98.62%	KDD'99
Sindhu et al. (2012)	IDS	Genetic Algorithm	DT	98.38%	KDD'99
Bolón-Canedo et al. (2011)	IDS	Correlation-based, Symmetrical Uncertainty, Consistency-based	SVM, MLP, DT: C4.5, Naive Bayes	94.08%	KDD'99
Baig et al. (2013)	IDS	Information Gain, Gain Ratio, Group method for data handling	ANN	97.72%	KDD'99
Zhang and Wang (2013)	IDS	SBS	Bayesian Network(BN)	98.98%	NSL-KDD
Ji et al. (2016)	IDS	analysis of variance	SVM, ANN, NB	96.67%	NSL-KDD
De La Hoz et al. (2014)	IDS	Multi-objective Optimisation Approach	Growing Hierarchical SOM	99.12%	NSL-KDD
De la Hoz et al. (2015)	IDS	PCA and FDR	Probabilistic SOM	94%	KDD'99
Koc et al. (2012)	IDS	Correlation-based, Consistency-based, INTERACT	NB	99.6%	KDD'99
Hosseini Bamakan et al. (2016)	IDS	Time-varying Chaos Particle Swarm Optimization	Multiple Criteria Linear Programming, SVM	97.84%	NSL-KDD
Singh et al. (2015)	IDS	Ensemble of filtered, correlation and consistency based feature selection	online sequential extreme learning machine	98.67%	NSL-KDD
Eesa et al. (2015)	IDS	Cuttlefish Algorithm	DT	91.986%	NSL-KDD
Kuang et al. (2014)	IDS	kernel PCA	SVM	96% (DR)	KDD'99

Table 3
Results of different topologies.

No.	Topo	Acc	Pre	DR	FAR
1	5-4-1	0.9869	0.9810	0.9933	0.0198
2	5-6-1	0.9867	0.9810	0.9930	0.0197
3	5-8-1	0.9817	0.9744	0.9898	0.0266
4	5-12-1	0.9860	0.9808	0.9921	0.0203
5	5-4-4-1	0.9886	0.9841	0.9936	0.0165
6	5-4-8-1	0.9846	0.9805	0.9893	0.0203
7	5-8-4-1	0.9841	0.9798	0.9890	0.0209
8	5-8-8-1	0.9857	0.9887	0.9933	0.0021

Table 4
Results of different features.

No.	Topo	Feature Selection	Acc	Pre	DR	FAR
9	4-4-1	remove dstPort	0.9822	0.9738	0.9915	0.0273
10	4-4-1	remove srcPort	0.9787	0.9690	0.9896	0.0324
11	4-4-1	remove srcIP	0.6785	0.7056	0.6263	0.2680
12	4-4-1	remove flag	0.6008	0.9969	0.2120	0.0007
13	4-4-1	remove sequence	0.9862	0.9811	0.9818	0.0196
14	2-4-1	only srcIP& flag	0.9785	0.9701	0.9879	0.0312

Table 5
Confusion matrix.

		Predicated	
		Attack	Normal
Actual	Attack	True Positive (TP)	False Negative(FN)
	Normal	False Positive(FP)	True Negative (TN)

ery feature from the original feature set to test the performance. The results show that removing srcIP or TCP flag leads to a sharp degradation on the performance, but on the contrary, removing the other three features does not cause significant changes. Therefore, it is reasonable to infer that the detector can still perform excellent performance with only using the srcIP and TCP flag as input features, which is proved by the result of the No.14 experiment. Besides, removing the "sequence" feature could even slightly improve the accuracy. So, we could point out that the original features used in paper Saied et al. (2016) have some redundant or irrelevant ones when they are used to detect the DDoS attack in ISOT data. Although our experiments are insufficient and ignore other factors that influence the performance of an MLP model, the results can still reveal that how well features characterize the traffic is a critical factor in determining the detection performance. Therefore, we think it is necessary to combine feature selection with the MLP model for constructing a well-performed detector, which is the first problem we need to solve.

In order to see how the performance changes when the traffic changes, we experiment with three datasets, which are denoted as D1, D2, and D3 (details are described in Section 5), to simulate the situation. All of them are synthesized by a nearly equal number of labeled normal and attack samples. The attack samples of every dataset are sampled from the same source, but the normal samples are sampled from three different sources. Dataset D1 is divided into two datasets to train the MLP model and be detected in phase 1. Dataset D2 and D3 are detected in phases 2 and 3. The results are shown in Table 6. We can see that the performance in phase 1 is as good as in the training phase, and the performance in phase 3 has a little degradation on Pre and FAR, but the performance in phase 2 has a prominent degradation on Acc, Pre and FAR. These results indicate that the detector, which has learned patterns characterized by five features from D1 train data, is well generalized on D1 and D3 test data but makes many errors on D2 test data, and according to the results of phase 2, these errors are mainly false-positive errors. Therefore, the tuned detector

Table 6
Results of different traffics.

Data Phase	D1(train) Train	D1(test) 1	D2(test) 2	D3(test) 3
Acc(%)	93.27	93.29	54.04	93.24
Pre(%)	89.38	89.47	52.06	88.26
DR(%)	98.21	98.12	99.68	99.72
FAR(%)	11.67	11.55	91.41	13.21

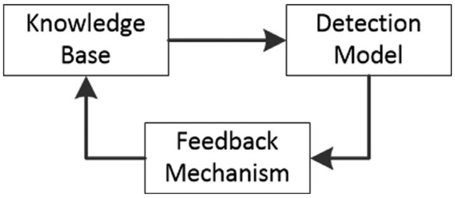


Fig. 2. Framework of the proposed method.

is possible to behave terrible detection performance and make a lot of errors (including false positive and negative errors) when traffic changes (including normal and attack traffic). Our experiments can reveal that performance degradation is because we use the labeled samples to compute the real performance metrics, but in practical detection, all samples that will be classified are unlabeled, and the following traffic is expected to be well generalized. However, this expectation is difficult to achieve because we do not know how the behavior of legitimate users changes and how DDoS attacks are launched and evolved. The hypothesis space in real detection is much bigger than the space in train data due to the unlimited changeable traffic. This point makes recognizing the DDoS attack a little different from a typical pattern recognition problem, such as handwriting recognition. Because the former is more subjective as DDoS attacks are usually a malicious abuse of normal packets, while the latter is more objective as the meaning of a character never changes. In order to ensure the availability of a tuned detector, we try to design a feedback mechanism to perceive the occurrence of many detection errors and then reconstruct the detector to avoid making more errors in the following detection.

4. Design of proposed method

In this section, we describe the dynamic MLP-based DDoS attack detection method. Our method has three modules: knowledge base, detection model, and feedback mechanism. A brief framework of the method is shown in Fig. 2.

4.1. Knowledge base

The knowledge base maintains two labeled datasets, including a training dataset and a feedback dataset, which are denoted as D_t and D_f. Dataset D_t is composed of the samples that are used to train the detection model, and D_f contains the new samples classified and labeled by the detector during detection. Every sample in these two data is an n-dimensional vector according to the definition of features. All samples are preprocessed to make them can be computed by the detection model, including transforming symbol features to numerical values, discretization, and normalization. To ensure the detection performance, the sample labels in D_t should be credible. How to prepare D_t belongs to another problem, and our work is based on some ready-made data. In practice, we can use a script to collect successfully served samples, use a honeypot to collect up-to-date DDoS attack traffic coupled with genuine application traffic, or resort to a credible third party. Dataset D_t is manually or automatically updated when the feedback mechanism triggers a signal.

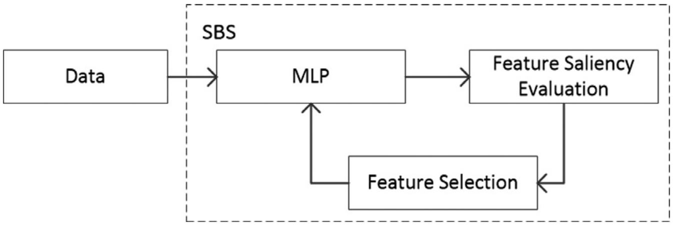


Fig. 3. Design of detection model.

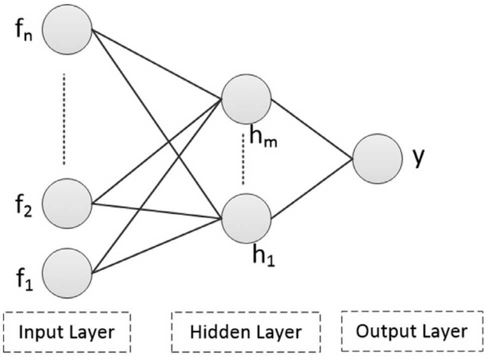


Fig. 4. Topology of MLP classifier.

When there are attack samples in current detection, D_f begins to record the new samples generated in a time window. As the traffic rate in a busy network is high-speed, we take a sampling strategy to record the new samples. The sampling rate of different samples is configured according to the corresponding generation rate. For example, when the traffic rate is slow, we can record all samples generated in the time window, and when the rate of normal samples is fast while the rate of attack samples is slow, we can use a low sampling rate to record normal samples while recording all attack samples. Every sample is tagged with a timestamp when it is added to D_f, and the samples out of the time window will be abandoned.

When we use D_t or D_f to train the MLP classifier, they are firstly divided into train and validation datasets according to a proper proportion. In order to avoid the sample imbalance problem, the proportion between attack and normal samples in the training data is also adjusted based on oversampling and undersampling techniques. For example, we can use the synthetic minority oversampling technique to generate synthetic minority samples.

4.2. Detection model

As we treat the DDoS attack detection as a binary classification problem, detection model is responsible for classifying the upcoming samples as normal or attack. In this paper, we use the MLP model as a classifier, and use a wrapper feature selection named SBS to select the optimal features. The design of the detection model is shown in Fig. 3. The topological structure of MLP is shown in Fig. 4. The neuron number of input layer is initially determined by the number of original features, and it will be changed during feature selection iteration procedure. The activation function used in the model is a logistic sigmoid function defined as

$$f(a) = \frac{1}{1 + exp(-a)}. \tag{5}$$

The output layer of the model has a single neuron to code the classification result according to the conditional probability $p(attack|x) = y$, with the probability $p(normal|x)$ given by $1 - y$. Given a training dataset comprising input vectors $\{\mathbf{x}_i\}$, where $i = 1, \ldots, N$, together with the corresponding target vectors $\{t_n\}$, we use the cross-entropy function defined in (6) as cost function (also called error function) (Nielsen, 2015).

$$E = -\frac{1}{N}\sum_{i=1}^{N}[t_i \ln y_i + (1-t_i)\ln(1-y_i)] \tag{6}$$

In order to reduce the computing time of the SBS procedure, the MLP model has only one hidden layer, and the neuron number of hidden layer is preset and keeps invariable during feature selection. The learning algorithm of gradient descent optimization, error backpropagation, and more details about MLP model can be seen in Bishop (2006). Supposing the original feature set is denoted as $F_0 = \{f_1, f_2, \ldots, f_n\}$ and MLP model is denoted as M, our goal is to find a subset $F^* \subseteq F_0$ that makes model M perform the best detection accuracy on data D_t^{test}. The proposed algorithm SBS-MLP is described as Algorithm 1.

Algorithm 1 SBS-MLP.

```
Require: $F_0, M, D_t^{train}, D_t^{validation}, D_t^{test}$
Ensure: $F^*, M, P_{cm}$
  $F_0 = \{f_1, f_2, \ldots, f_n\}, F' = \emptyset, F_1 = F_0$
  Train $M$ on $D_t^{train}$ and $D_t^{validation}$ with the features in $F_1$ as inputs
  Test the trained $M$ on $D_t^{test}$ to get the feature saliency $S_{(1,\emptyset)} = 1 - accuracy$
  $C_{F_1} = S_{(1,\emptyset)}$
  for $i = 1 \rightarrow n-1$ do
    for each $f \in F_i$ : do
      $H = F_i - f$
      Train $M$ on $D_t^{train}$ and $D_t^{validation}$ with the features in $H$ as inputs
      Test the trained $M$ on $D_t^{test}$ to get the feature saliency $S_{(i,f)} = 1 - accuracy$
    end for
    $f^* = \arg\min_f S_{(i,f)}$
    $F_{i+1} = F_i - f^*$
    $C_{F_{i+1}} = \min S_{(i,f)}$
  end for
  $F^* = \arg\min_{F_i} |F_i|$ subject to $\max(C_{F_i}) - C_{F_i} \leqslant \varepsilon$
  Train $M$ on $D_t^{train}$ and $D_t^{validation}$ with the features in $F^*$ as inputs
  Test the trained $M$ on $D_t^{test}$ to get the confusion matrix $P_{cm}$
  Return $F^*$, $M$, and $P_{cm}$
```

From the SBS-MLP algorithm, we could see that it is a top-down procedure. This procedure starts with the full set F_0 to train and test the MLP model. The detection accuracy on test data and the corresponding features are recorded as the first candidate (i.e., C_{F_1}). Then, every feature in the current feature set is temporally removed to compute feature saliency in the internal iteration. Feature saliency determines the criterion of evaluating the importance of a feature. In order to emphasize the generalization performance, we use $1 - accuracy$ on test data as the function to evaluate the saliency. This function can also be replaced by other functions, such as the combination function of multi metrics and sum-of-squares error (SSE) loss function. After the internal iteration procedure is finished, the feature with the minimal feature saliency is permanently removed from the feature set, and the feature set that has the maximal accuracy is recorded as the next candidate. After the external iteration procedure is finished, we find the optimal feature subset according to a constrained optimization problem, which is defined as

$$F^* = \arg\min_{F_i} |F_i| s.t. \max(C_{F_i}) - C_{F_i} \leqslant \varepsilon, (i = 1, 2, \ldots, n). \tag{7}$$

According to (7), we choose the subset which has the minimal cardinality from all candidates that are close enough to the best detection accuracy as the optimal feature subset. Parameter ε controls the tradeoff between better detection accuracy and less number of features, and it should be set to a small non-negative value to ensure the priority of detection accuracy. For example, we probably prefer to choose a feature subset with fewer features when its accuracy is only a few tenths of a percentage point less than the maximal accuracy. After the optimal feature subset is selected, the MLP model is retrained with the features in the subset as inputs. Then, the tuned model is tested on test data to get the confusion matrix, which will be used in the feedback mechanism. In practice, there is no need to train and test the model again, because we can save model parameters and corresponding results during the iteration procedure, but for the brevity of the algorithm, here we use a redundant description. Finally, the detection model M can be deployed to detect genuine traffic.

According to the iteration procedure, the ith iteration will train and test the model n-i+1 times, so the total times are $1 + \sum_{i=1}^{n-1}(n-i+1) = n(n+1)/2$. Compared with the complexity of using brute force search, SBS strategy reduces the search space from $O(2^n)$ to $O(n^2)$. In addition, as the internal iteration procedure can be executed concurrently, the complexity can be further reduced to $O(n)$. This reduction of computing complexity makes it feasible to retrain the model before evaluating the feature saliency of a temporarily removed feature when the number of original features is too large, which is recommended by paper Romero and Sopena (2008). However, a disadvantage of the SBS search strategy is that it is not a greedy selection. Because the optimal solution of the problem does not always contain the optimal solution of the sub-problem, as features can interact with each other in a non-linear and complex way. More specifically, only the first iteration can search in all C(n,n-1) different feature combinations to find the optimal solution for the n-1 dimension sub-problem, but the ith $(i \neq 1)$ iteration can only search in the C(n-i,n-i-1) different combinations without considering the other [C(n,n-i-1)-C(n-i,n-i-1)] combinations to find the optimal solution for next sub-problem.

4.3. Feedback mechanism

The feedback mechanism is responsible for perceiving the occurrence of considerable detection errors based on the new labeled samples recorded in D_f. It is executed only when the number (or proportion) of the new labeled attack samples in D_f (denoted as N_a) exceeds a certain value (denoted as N_0) because enough number of attack samples is necessary. The basic hypothesis of the mechanism is that: when the number of false-positive/negative errors in current detection is accumulated to a certain degree, and if we use the new labeled samples in this duration to retrain the detection model, the detection accuracy of the retrained model on test data will appear distinguishable reduction compared with the original model. On the contrary, if the detector is well-performed, the accuracy reduction should only fluctuate in a limited range. Therefore, we can determine whether the detector is making considerable errors by comparing the accuracy reduction with a preset threshold, which represents the normal accuracy reduction profile under the stable condition when the detector performs well. If current accuracy reduction is bigger than the threshold, the feedback mechanism will trigger a signal to update D_t and reconstruct the detection model. The proposed error perceiving algorithm (EPA) is described as follows:

In Algorithm 2, parameter θ is the important threshold used to make decisions, and it is computed based on the Bienaymé-Chebyshev's inequality, which is defined as:

$$P(|\delta_s - \mu| \geqslant k\sigma) \leqslant \frac{1}{k^2}. \tag{8}$$

Algorithm 2 Error perceiving algorithm.

```
while $N_a \geq N_0$ do
    Read data from $D_f$
    Train $M$ on $D_f^{train}$ and $D_f^{validation}$ with the features in $F^*$ as inputs
    Test the trained $M$ on $D_t^{test}$ to get confusion matrix $Q_{cm}$
    Compute detection accuracy $a_P$ and $a_Q$ according to $P_{cm}$ and $Q_{cm}$
    $\delta = a_P - a_Q$
    if $\delta > \theta$ then
        Update $D_t$
        Execute SBS-MLP procedure with new $D_t$
        Update $\theta$
    end if
end while
```

The inequality can be used for outliers detection when data distribution is unknown De Assis et al. (2018). δ_s is a random variable and denotes the normal accuracy reduction under stable traffic condition, μ is the mean value, σ is the standard deviation, and $k > 0$ is the deviation parameter. Assuming we want the usual cut-off point for statistical significance to be 0.05, we can get $k = 4.47$ and compute the threshold θ accordingly. In case that a value of reduction is higher than θ, then we have 95% confidence level to consider this point as anomalous. The unbiased estimators of μ and σ are used to compute the threshold as follows:

$$\widehat{\mu} = E(\delta_s) = \frac{1}{N}\sum_{i=1}^{N}\delta_s(i), \tag{9}$$

$$\widehat{\sigma} = \sigma(\delta_s) = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}[\delta_s(i) - \mu]^2}, \tag{10}$$

$$\theta = \widehat{\mu} + k\widehat{\sigma}. \tag{11}$$

As we only concern the abnormal increase of the accuracy reduction, we only use the upper bound of the inequality. Every time the reconstruction is executed, the threshold θ will be updated according to the equations. When the threshold is triggered, the process of feedback mechanism is suspended until the reconstruction is finished.

5. Experiment and discussion

In this section, we firstly describe the experiments about validating the effectiveness of the detection model and feedback mechanism and then give a discussion about our work. All experiments were operated on the Matlab R2014a platform. The server used to run the program had 8 CPUs of i7-4790 with 3.60 GHz and 32 GB RAM.

5.1. Experiment of detection model

Firstly, we implemented a detection model with the topological structure implemented in Section 3 to validate the SBS-MLP algorithm. We carefully collected the attack and normal packets from dataset ISOT, the normal packets from dataset ISCX (Shiravi et al., 2012), and the custom normal packets captured from our campus network. Then we relabeled all packets and synthesized them as shown in Table 7. When we run the program, the train data was automatically divided into train, validation, and test dataset with the proportion of 70%:15%:15%. The training phase stopped at where a minimum cross-entropy on the validation dataset was achieved. To better test the generalization performance, we only used the detection accuracy on the test data in Table 7 as the final results.

Table 7
Composition of experiment data.

Data	Combination	Train (No. of packets)	Test (No. of packets)
D1	ISOT Attack	120,000	129,455
	ISOT Normal	120,000	129,457
D2	ISOT Attack	120,000	129,455
	ISCX Normal	120,000	130,000
D3	ISOT Attack	120,000	129,455
	Custom Normal	120,000	129,973

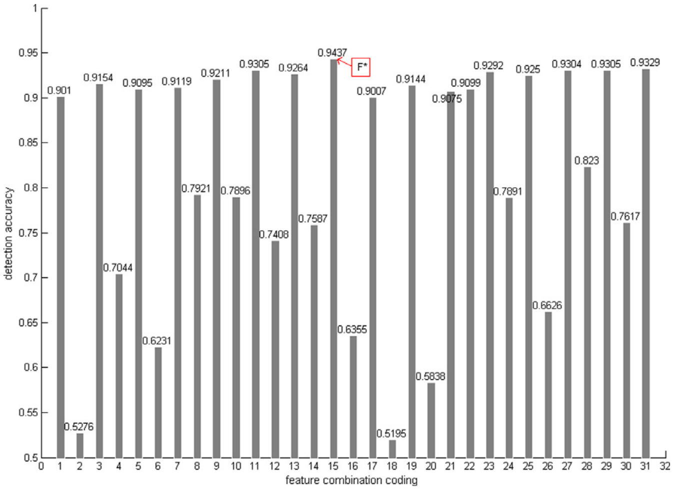


Fig. 5. Results of brute force search on D1 (5 features). F^* is the optimal feature subset encoded as 11110=15.

According to Fig. 1, the original feature set F_0 has five features, and they are directly extracted from the packet header field, including source IP, TCP sequence, TCP flag, and TCP source and destination port. For brevity, all subsets of F_0 ({srcIP, srcPort, dstPort, tcpSeq, tcpFlag}) are encoded through a five bits binary number and transformed to a decimal value. The bit assigned with 1 indicates the corresponding feature is selected, and 0 means not. For example, value 19 equals to 11001 and indicates only srcIP, srcPort, and tcpFlag are selected as the input features. We also compared SBS-MLP with the other two algorithms that combined the sequential selection techniques with MLP, including SFS and clamping technique based SBS (CTSBS), which were first proposed in Vesa et al. (2001) and Baesens et al. (2000) respectively. The results of D1 in every external iteration procedure are shown in Table 8, and the results of D2 and D3 are shown in Table 9 and Table 10. As five features only have $2^5 - 1 = 31$ different combinations, it is feasible to test all possible feature subsets through brute force search. The results are shown in Fig. 5, where the horizontal axis represents the decimal encoding of all subsets from 1 to 31. In order to further evaluate the SBS-MLP algorithm, we enlarged the feature set to 12 features and tested the three algorithms on D1, D2, and D3. The results are shown in Table 11. The 12 features are directly extracted from the packet header filed, including total length, identification, flags, time to live, source and destination IP address, TCP source and destination port, TCP sequence, TCP ACK, TCP flags, and TCP window (they are numbered serially from 1 to 12 in Table 11). Parameter ε in all experiments is set to zero, which means we give top priority to the detection accuracy.

According to the results in Tables 8–11, we find that: (1) SBS-MLP can find the optimal feature subset that has fewer features and performs better accuracy than the original full feature set; (2) the optimal subset given by SBS-MLP produces better detection performance than CTSBS-MLP and SFS-MLP with considering the four metrics; (3) the optimal feature subsets of different data

Table 8
Experiment results of D1 (5 features).

Iteration					1				2				3				4				5			
SBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	93.29	89.47	98.12	11.55	94.37	90.74	98.84	10.09	93.05	89.01	98.22	12.12	92.11	88.5	96.79	12.58	90.1	84.88	97.59	17.38
	features				11111(31)(F_0)				11110(15)(F^*)				11010(11)				10010(9)				10000(1)			
SFS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	90.1	84.88	97.59	17.38	92.11	88.5	96.79	12.58	93.05	89.01	98.22	12.12	94.37	90.74	98.84	10.09	93.29	89.47	98.12	11.55
	features				10000(1)				10010(9)				11010(11)				11110(15)(F^*)				11111(31)(F_0)			
CTSBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	93.29	89.47	98.12	11.55	94.37	90.74	98.84	10.09	92.64	88.69	97.75	12.47	74.08	65.87	99.93	51.77	79.21	70.64	99.98	41.55
	features				11111(31)(F_0)				11110(15)(F^*)				10110(13)				00110(12)				00010(8)			

Table 9
Experiment results of D2 (5 features).

Iteration					1				2				3				4				5			
SBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	82.44	89.55	73.37	8.53	89.35	90.92	87.38	8.69	88.26	90.54	85.4	8.89	87.73	89.3	85.67	10.22	84.97	85.04	84.79	14.85
	features				11111(F_0)				11101(F^*)				10101				10001				10000			
SFS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	84.97	85.04	84.79	14.85	87.73	89.3	85.67	10.22	88.89	89.25	88.38	10.6	89.1	89.55	88.47	10.28	82.44	89.55	73.37	8.53
	features				10000				10001				10011				11011(F^*)				11111(F_0)			
CTSBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	82.44	89.44	73.37	8.53	89.0	90.74	86.82	8.83	88.89	89.25	88.38	10.6	80.76	87.68	71.47	10.0	84.97	85.04	84.79	14.85
	features				11111(F_0)				10111(F^*)				10011				10010				10000			

Table 10
Experiment results of D3 (5 features).

Iteration					1				2				3				4				5			
SBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	87.57	80.11	99.89	24.71	97.83	96.49	99.26	3.59	96.43	94.43	98.67	5.8	97.88	96.61	99.23	3.47	91.87	88.22	96.6	12.85
	features				11111(F_0)				11110				11100				10100(F^*)				10000			
SFS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	91.87	88.22	96.6	12.85	97.88	96.61	99.23	3.47	96.43	94.43	98.67	5.8	97.83	96.49	99.26	3.59	87.57	80.11	99.89	24.71
	features				10000				10100(F^*)				11100				11110				11111(F_0)			
CTSBS-MLP	Acc(%)	Pre(%)	DR(%)	FAR(%)	87.57	80.11	99.89	24.71	81.34	72.8	99.95	37.21	55.4	52.81	99.98	88.99	50.52	50.21	99.96	98.73	48.56	49.21	96.07	98.76
	features				11111(F_0, F^*)				11101				10101				10001				00001			

are different with each other even when the original features are same. The first point indicates that the proposed SBS-MLP algorithm is effective to improve the detection accuracy through feature selection. The second point indicates that SBS-MLP performs better than the other two algorithms. The third point indicates that the optimal feature subset changes when the traffic changes. Besides, combining the results in Fig. 5 with Table 8, we find that SBS-MLP and SFS-MLP both can get the global optimal feature subset (F^* is 11110=15). We also have tested the brute force search on D2 and D3, and the results indicate that the optimal feature subsets of D2 and D3 given by SBS-MLP and SFS-MLP are global as well. However, when the number of features is large, whether the optimal subset is a global solution is uncertain, but a suboptimal solution is still acceptable. From Table 11, we could see that all accuracies corresponding to the full feature set and optimal feature subset are improved to a very high level after we enlarged the number of features from 5 to 12. As we can adjust parameter ε to control the tradeoff consideration between detection accuracy and feature number, SFS-MLP may behave better than SBS-MLP. In this situation, we can run the SBS-MLP and SFS-MLP procedure together if it is feasible and then assemble their results to select the optimal feature subset according to (7).

At last, in order to compare our method with some related works using the same data, we also did experiments on the NSL-KDD dataset (Mahbod, 2009), which is a widely used benchmark data in the area of intrusion detection. This dataset has five main categories: normal, probing, denial of service (DoS), user to root (U2R), and remote to local (R2L), and it has 41 features. As we only concern about DDoS attacks, we extracted all normal and DoS attack samples to organize a new train and test dataset (as shown in Table 12). Then we did experiments on the reorganized datasets. In the experiments, the number of hidden layer neurons was added to twenty-two. All results are shown in Table 13. As many referenced works only have Acc, DR, and FAR metrics, here we only list these three metrics in the table. We pick out the results of detecting the DoS attack and compare it with them if there are the details, but if the referenced works only have the holistic results, we label the corresponding comparisons with a “*” symbol. According to Table 13, although the detection performance of our proposed SBS-MLP is not the best, it still performs a comparable performance with 97.66% Acc, 94.88% DR, and 0.62% FAR. Taking the three metrics into consideration, the SBS-MLP can be regarded as performing the best performance among all MLP-based detection methods. In addition, although the number of features (NoF) of the optimal subset given by SBS-MLP is 31, our method can also find an optimal subset that has 16 features and produces 97.29% Acc, 94.86% DR and 1.2% FAR after we increased the tradeoff parameter ε.

5.2. Experiment of feedback mechanism

We used the detection model with 12 features implemented in the last section to validate the feedback mechanism. After the detection model was tuned by the D1 dataset (i.e. D_t), we could get the best test accuracy $a_P = 99.62\%$ (see Table 11). To see how the accuracy reduction was influenced, we used three groups of data as the feedback dataset D_f, which are denoted as G1, G2, and G3 (see Table 14), to simulate three different situations through gradually increasing the proportion of new normal packets in traffic. More concretely, G1 simulates the stable situation that current traffic is stable and all packets come from the ISOT dataset. G2 and G3 simulate two different situations that current traffic has more and more new packets came from the ISCX traffic and the custom traffic. In the experiments, we set the total sample number of D_f in a time window as 40,000 samples while keeping the proportion between attack and normal samples around fifty-fifty. The attack

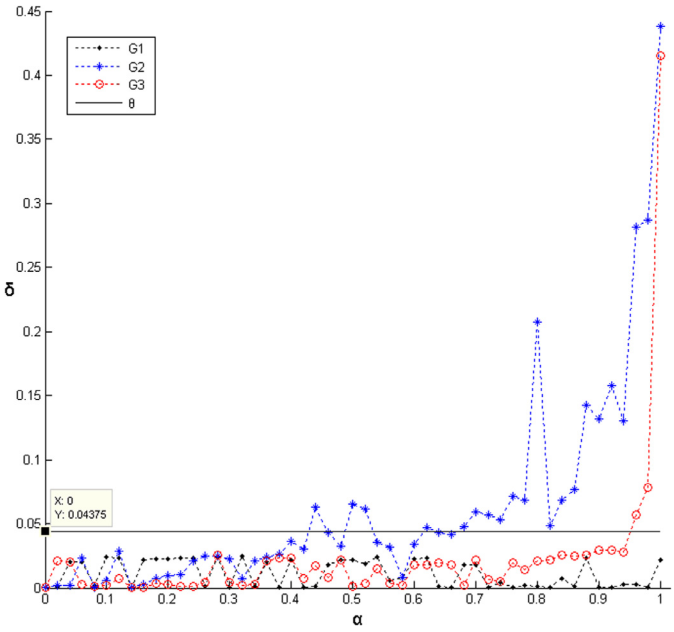


Fig. 6. Accuracy reduction of G1, G2, and G3.

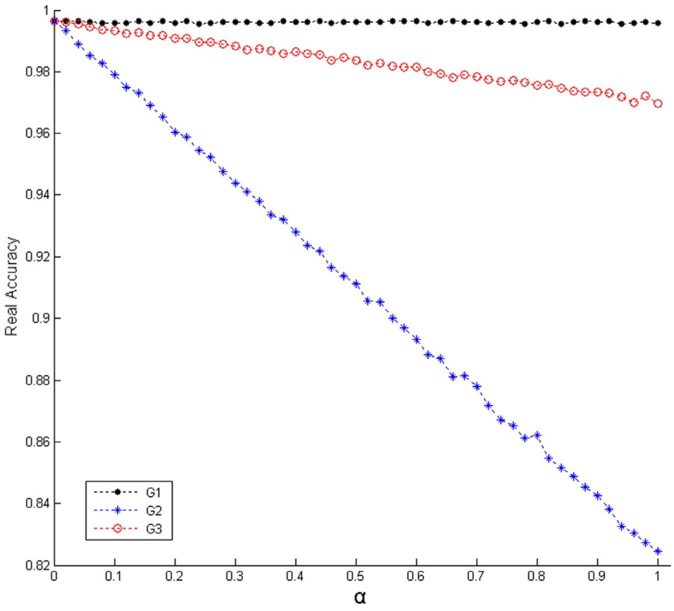


Fig. 7. Real accuracy of G1, G2, and G3.

samples of each dataset are randomly sampled from the ISOT test dataset. The normal samples of each dataset are randomly sampled from the corresponding test dataset and synthesized according to proportion α. Parameter α controls the proportion between ISOT normal packets and other new normal packets in each dataset, and it is increased from 0 to 1 with 0.02 increment every step. So, we could get 51 different D_f datasets in each group of data, and then we serially run the EPA procedure on every D_f to get the values of δ. The results are shown in Fig. 6. As the samples in all datasets are labeled, we can also compute the real accuracy at each step to see the how the detector degrades with the proportion of new normal packets increasing(see Fig. 7).

From Fig. 6, we can see that the accuracy reduction δ of G1 (black point) only fluctuates in a small range with the parameter α increasing, which is consistent with our hypothesis. As we use G1 to simulate the stable situation, the δ values of G1 depict the normal profile of the accuracy reduction, and we can compute the threshold accordingly. Parameter μ and σ are firstly estimated according to (9) and (10), and by setting the confidence level k in (11) as 0.1, we can get the threshold $\theta = 0.04375$ (black line in Fig. 6). Then we can see that the δ of G2 (blue star) firstly exceeds the threshold at $\alpha = 0.44$ and continuously exceeds the threshold after α is bigger than 0.68, and the δ of G3 (red circle) exceeds the threshold only when α is bigger than 0.96. According to the EPA procedure, a reconstruction of the detection model will be triggered when the accuracy reduction exceeds the threshold. The changes of real accuracy shown in Fig. 7 demonstrates that the reconstruction is advisable, and the feedback mechanism is effective. From Fig. 7, we can see the real accuracies of G2 and G3 both decrease with the proportion of new normal packets increasing, which indicates that the detector is making more and more errors. Meanwhile, the real accuracy of G2 decreasing much steeper than G3 indicates that the new normal packets from ISCX traffic impact much more on the detection performance than the custom traffic dose, in other words, the detector is better generalized on the traffic captured from our campus network than the ISCX traffic, which conforms to the results in Section 3. Besides, the difference in real accuracy between G2 and G3 in Fig. 7 is also consistent with the difference in accuracy reduction shown in Fig. 6. The feedback mechanism responding much sooner on G2 than it dose on G3 is because the degradation of real accuracy on G2 is much worse than G3. In summary, these results support the basic hypothesis we proposed in the feedback mechanism, and the EPA procedure is effective to perceive the errors made by the detector due to the not well-generalized traffic.

As the results of G2 have more significance in showing the impact on detection accuracy when traffic changes, we further test the accuracy reduction and real accuracy of G2 after the detector is reconstructed to see how it is improved. Supposing the reconstruction procedure is triggered at α= 0.44, the detection model is retrained through an up-to-date dataset D_t (here we directly updated the dataset through synthesizing the samples in D1 and D2 training datasets). After the procedure is finished, new accuracy reduction and real accuracy of G2 are computed, and the new stable situation is also simulated in the same way as G1. The results are shown in Figs. 8 and 9. In Fig. 8, the black points represent the accuracy reduction of new stable traffic, and we can see it still only fluctuates in a small range. The new threshold θ (brown line) is computed accordingly. The brown forked points started from the vertical dashed line, where α= 0.44, shows the changes in accuracy reduction of G2 after reconstruction. We can see that the accuracy reduction is under the threshold again, and it fluctuates under the threshold within a certain interval of α. Meanwhile, the optimal feature subset changes from (1,2,4,5,6,7,9,12) to (1,3,4,5,6,7,9,10,11,12), and the real accuracy of G2 returns from about 92% to more than 99% (as shown in Fig. 9). These results demonstrate that the feedback mechanism is effective to correct the detector when it is making considerable errors.

At last, we should pay attention to two problems. First, as shown in Fig. 6, there are some inconsistent points of G2 that lead to some failures around α= 0.58 when the detector performs poorly. Although the failure sometimes occurs, the detection model and feedback mechanism can still work well, because this kind of false-negative response only occurs when the performance degradation is tolerable and will be remedied in the following execution cycles through the next D_f. Second, as shown in Fig. 8, the δ of G2 after reconstruction exceeds the new threshold again seven times after the proportion α is bigger than 0.7 while the new detector still performs a very high detection accuracy, which will lead to some false-positive responses. According to our observation, this is because the normal samples in D_f gradually changes from the

Table 11
Experiment results of D1, D2, and D3 (12 features).

Data		D1		D2		D3	
Feature Set		F_0	F^*	F_0	F^*	F_0	F^*
SBS-MLP	Acc(%)	99.4	99.62	98.68	99.99	90.71	99.99
	Pre(%)	99.2	99.59	99.56	99.99	84.34	99.98
	DR(%)	99.61	99.65	97.78	99.99	99.94	99.99
	FAR(%)	0.8	0.41	0.43	0.01	18.49	0.02
	features	all	1,3,4,5,6,7,9,10,11,12	all	4,5,6,8,11,12	all	5,6,7,8,9,10,12
SFS-MLP	Acc(%)	99.4	99.4	98.68	99.98	90.71	99.99
	Pre(%)	99.2	99.2	99.56	1.00	84.34	99.92
	DR(%)	99.61	99.61	97.78	99.97	99.94	99.92
	FAR(%)	0.8	0.8	0.43	0.00	18.49	0.01
	features	all	all	all	4,5,6,7,8,9,10,11,12	all	3,5,6,7,8
CTSBS-MLP	Acc(%)	99.4	99.48	98.68	99.99	90.71	95.35
	Pre(%)	99.2	99.35	99.56	99.99	84.34	91.9
	DR(%)	99.61	99.6	97.78	99.99	99.94	99.44
	FAR(%)	0.8	0.65	0.43	0.01	18.49	8.73
	features	all	1,3,4,5,6,9,10,12	all	4,5,6,8,10,11,12	all	1,2,3,4,5,7,11,12

Table 12
Composition of reorganized NSL-KDD dataset.

	Train	Test
number of normal	67,343	9711
number of DoS	45,927	6034

full ISOT normal packets to full ISCX normal packets, and when we retrain the detector through the new D_f, the dominant ISCX normal patterns make the retrained detector specific and can not be well generalized on the ISOT normal patterns. The false-positive response under this situation only wastes some computing resources and will not impact the detection performance.

5.3. Discussion

In the research community of detecting DDoS attacks, most researchers treat the detection as a classification problem and want to solve the problem once for all based on machine learning algorithms. However, many solutions are prone to becoming fallible or even invalid when they are deployed in the practical network due to the disabled features and the changeable traffic, although they have a very high detection accuracy on a given test data. In order to solve the problem, we try to build a closed-loop system by introducing a feedback mechanism to perceive detection errors. As a tentative intention, we proposed an MLP-based method to validate our idea, and the experiment results indicated that the idea is effective. Although our work is restricted to the MLP model, we think the proposed problem and the basic feedback framework can be extended to many other machine learning-based detection methods.

Considering the parallelism of ANN and the mature open-source implementation platform like Tensorflow, as well as the linear complexity of sequential feature selection, we think the proposed solution has good scalability. Besides, as the sampling rate of the feedback dataset can be adjusted and the feature statistic can be collected through the sampling flow (sFlow) technique (Lu and Wang, 2016), the feedback process is competent to work in a busy network in theory, but this needs further assessment in a more

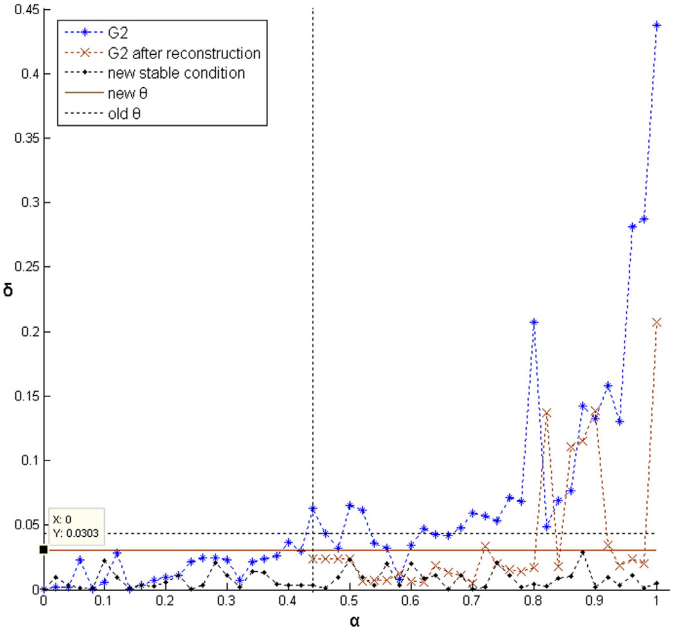


Fig. 8. Accuracy reduction of G2 after reconstruction.

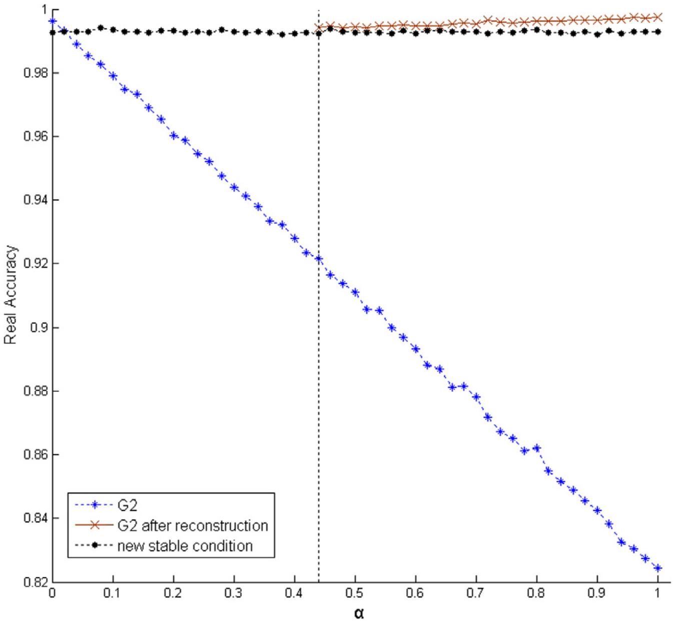


Fig. 9. Real accuracy of G2 after reconstruction.

Table 13
Comparison results on NSL-KDD.

Work	Detection Model	FS	NoF	Acc(%)	DR(%)	FAR(%)
SBS-MLP	MLP	Yes	31	97.66	94.88	0.62
SFS-MLP	MLP	Yes	35	97.61	94.71	0.6
CTSBS-MLP	MLP	Yes	41	97.61	94.78	0.63
Yusof et al. (2018)*	MLP	Yes	17	91.7	–	–
Hosseini and Azizi (2019)*	MLP	Yes	20	96.1	–	–
Osanaiye et al. (2016)*	DT:J48	Yes	13	99.67	99.76	0.42
Zhang and Wang (2013)*	BN	Yes	11	98.98	–	–
Ji et al. (2016)*	MLP	Yes	22	96.67	–	–
Kim and Gofman (2018)*	MLP	No	41	98.5	–	1.4
Kushwah and Ali (2018)*	MLP	No	41	96.3	94.37	5.0
Chuan-long et al. (2017)	RNN	No	41	93.42	86.3	0.93
De La Hoz et al. (2014)*	GH-SOM	Yes	25	99.12	–	2.24
Hosseini Bamakan et al. (2016)	SVM	Yes	17	99.18	98.98	0.69
	MCLP	Yes	17	98.26	98.68	2.03
Sabar et al. (2018)*	HH-SVM	No	41	85.69	–	–

Table 14
Composition of synthetic dataset.

	ISOT Attack	ISOT Normal	ISCX Normal	Custom Normal
G1	20,000	20,000	0	0
G2		(1-α)*20,000	α*20,000	0
G3		(1-α)*20,000	0	α*20000

comprehensive way. In our experiments, as the MLP structure used in the experiments was simple and the sample number of D_f was fixed to 40,000, the feedback process only took about 7 s to run the EPA procedure once, and reconstructing the detector through SBS-MLP algorithm took about 160 s (about 240,000 train samples) while the model training procedure only used about 12%–15% CPU resources due to the limitation of Matlab. We plan to implement our solution as a northbound application under the software-defined networking (SDN) paradigm, which supports implementing our solution based on the abstracted APIs without considering the low-level details (a similar work can be seen in Bawany et al. (2017). In this way, our method can work as an application software running on a cloud server, and the computing resources required can be allocated elastically according to the real-time overhead. How to realize the optimal resource allocation and scalability can refer to the works in Al-Haidari et al. (2013) and Calyam et al. (2014).

6. Conclusion

In this paper, one of our main purposes is to improve the availability of many modern machine learning-based detection methods, at least from the perspective of MLP model, as they seldom consider the unavailability in practical deployment caused by disabled features and changeable traffic. We demonstrated this problem, which is not widely addressed by the community, through implementing and testing an MLP-based case. In order to solve the problem, we proposed a dynamical MLP-based detection method against the DDoS attack through combing with sequential feature selection and feedback mechanism. According to the results, on the one hand, our method had comparable detection performance on the popular benchmark data NSL-KDD compared with some related works. On the other hand, our method was effective in perceiving the detection errors when their saliency accumulated to a certain degree and then reconstruct the detector according to updated data. The main contributions of our work include: (1) proposed an easily feasible and interactive method to combine feature selection with MLP model; (2) designed a feedback mechanism to perceive detection errors based on the recent detection results. The disadvantages of our method mainly include: (1) the SBS-MLP algorithm can not ensure finding the global optimal features, but a suboptimal solution is also acceptable; (2) the feedback mechanism is possible to generate false-negative or false-positive responses, but the former can be remedied by itself, and the latter does not impact the detection performance.

In future work, we aim to investigate a more effective and lightweight solution to perceive the detection errors, implement our solution based on the northbound API provided by open-source controller Opendaylight, and validate the solution in the SDN environment on our private cloud platform. We hope the planned works can extend our solution to meet the diverse and changeable application-specific DDoS attack detection requirements in the cloud environment.

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 is supported by the Research and Development Program in Key Areas of Guangdong Province, Novel Network Architecture and Key Technologies, including project No. 2018B010113001 and No. 2019B010137001.

References

Akilandeswari, V., Shalinie, S.M., 2012. Probabilistic neural network based attack traffic classification. In: Proceedings of the Fourth International Conference on Advanced Computing, ICoAC 2012 doi:10.1109/ICoAC.2012.6416848.

Al-Haidari, F., Sqalli, M., Salah, K., 2013. Impact of CPU utilization thresholds and scaling size on autoscaling cloud resources. In: Proceedings of the International Conference on Cloud Computing Technology and Science, CloudCom, 2, pp. 256–261. doi:10.1109/CloudCom.2013.142.

Baesens, B., Viaene, S., Vanthienen, J., Dedene, G., 2000. Wrapped feature selection by means of guided neural network optimisation. In: Proceedings 15th International Conference on Pattern Recognition. ICPR-2000, 2, pp. 113–116. doi:10.1109/ICPR.2000.906029.

Baig, Z.A., Sait, S.M., Shaheen, A., 2013. GMDH-based networks for intelligent intrusion detection. Eng. Appl. Artif. Intell. 26 (7), 1731–1740. doi:10.1016/j.engappai.2013.03.008.

Balkanli, E., Nur Zincir-Heywood, A., Heywood, M.I., 2015. Feature selection for robust backscatter DDoS detection. In: Proceedings of the Conference on Local Computer Networks, LCN, 2015-December, pp. 611–618. doi:10.1109/LCNW.2015.7365905.

Barati, M., Abdullah, A., Udzir, N.I., 2014. Distributed denial of service detection using hybrid machine learning technique. 2014 International Symposium on Biometrics and Security Technologies (ISBAST) 268–273. doi:10.1109/ISBAST.2014.7013133.

Bawany, N.Z., Shamsi, J.A., Salah, K., 2017. DDoS attack detection and mitigation using SDN: methods, practices, and solutions. Arab. J. Sci. Eng. 42, 425–441. doi:10.1007/s13369-017-2414-5.

Beigi, E.B., Jazi, H.H., Stakhanova, N., Ghorbani, A.A., 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In: Proceedings of the IEEE Conference on Communications and Network Security, CNS 2014, pp. 247–255. doi:10.1109/CNS.2014.6997492.

Bishop, C.M., 2006. Neural networks. In: Pattern Recognition and Machine Learning. Springer, New York, pp. 225–247. Chapter 5.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2011. Feature selection and classification in multiple class datasets: an application to KDD cup 99 dataset. Expert Syst. Appl. 38 (5), 5947–5957. doi:10.1016/j.eswa.2010.11.028.

Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A., 2013. A review of feature selection methods on synthetic data. Knowl. Inf. Syst. 34 (3), 483–519. doi:10.1007/s10115-012-0487-8.

Braga, R., Mota, E., Passito, A., 2010. Lightweight DDoS flooding attack detection using NOX/OpenFlow. In: IEEE Local Computer Network Conference, pp. 408–415. doi:10.1109/LCN.2010.5735752.

Calyam, P., Rajagopalan, S., Seetharam, S., Selvadhurai, A., Salah, K., Ramnath, R., 2014. VDC-Analyst: design and verification of virtual desktop cloud resource allocations. Comput. Netw. 68, 110–122. doi:10.1016/j.comnet.2014.02.022.

Chang-Jung, H., Ting-Yuan, C., 2016. Detection DDoS attacks based on neural-network using Apache Spark. In: 2016 International Conference on Applied System Innovation (ICASI), pp. 1–4. doi:10.1109/ICASI.2016.7539833.

Chen, J.H., Zhong, M., Chen, F.J., Zhang, A.D., 2012. DDoS defense system with turing test and neural network. In: Proceedings of the IEEE International Conference on Granular Computing, GrC 2012, pp. 38–43. doi:10.1109/GrC.2012.6468680.

Chen, X.F., Yu, S.Z., 2016. CIPA: a collaborative intrusion prevention architecture for programmable network and SDN. Comput. Secur. 58, 1–19. doi:10.1016/j.cose.2015.11.008.

Chuan-long, Y., Yue-fei, Z., Jin-long, F., Xin-zheng, H., 2017. A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access 3536 (c), 1. doi:10.1109/ACCESS.2017.2762418.

Criscuolo, P.J., 2000. Distributed Denial of Service, Tribe Flood Network 2000, and Stacheldraht CIAC-2319. Technical Report. Lawrence Livermore National Laboratory.

Cui, Y., Yan, L., Li, S., Xing, H., Pan, W., Zhu, J., Zheng, X., 2016. Sd-anti-DDoS: fast and efficient DDoS defense in software-defined networks. J. Netw. Comput. Appl. 68, 65–79. doi:10.1016/j.jnca.2016.04.005.

De Assis, M.V., Novaes, M.P., Zerbini, C.B., Carvalho, L.F., Abrao, T., Proenca, M.L., 2018. Fast defense system against attacks in software defined networks. IEEE Access 6, 69620–69639. doi:10.1109/ACCESS.2018.2878576.

De La Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., Martínez-Álvarez, A., 2014. Feature selection by multi-objective optimisation: application to network anomaly detection by hierarchical self-organising maps. Knowl. Based Syst. 71, 322–338. doi:10.1016/j.knosys.2014.08.013.

Eesa, A.S., Orman, Z., Brifcani, A.M.A., 2015. A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert Syst. Appl. 42 (5), 2670–2679. doi:10.1016/j.eswa.2014.11.009.

Hosseini, S., Azizi, M., 2019. The hybrid technique for DDoS detection with supervised learning algorithms. Comput. Netw. 158, 35–45. doi:10.1016/j.comnet.2019.04.027.

Hosseini Bamakan, S.M., Wang, H., Yingjie, T., Shi, Y., 2016. An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing 199. doi:10.1016/j.neucom.2016.03.031.

De la Hoz, E., De La Hoz, E., Ortiz, A., Ortega, J., Prieto, B., 2015. PCA filtering and probabilistic SOM for network intrusion detection. Neurocomputing 164, 71–81. doi:10.1016/j.neucom.2014.09.083.

Inc., A.N., 2018. NETSCOUT Arbor's 13th Annual Worldwide Infrastructure Security Report. Technical Report.

Ji, S.Y., Jeong, B.K., Choi, S., Jeong, D.H., 2016. A multi-level intrusion detection method for abnormal network behaviors. J. Netw. Comput. Appl. 62, 9–17. doi:10.1016/j.jnca.2015.12.004.

Khan, M.A., Salah, K., 2018. IoT security: review, blockchain solutions, and open challenges. Future Gener. Comput. Syst. 82, 395–411. doi:10.1016/j.future.2017.11.022.

Kim, D.E., Gofman, M., 2018. Comparison of shallow and deep neural networks for network intrusion detection. In: Proceedings of the IEEE Eighth Annual Computing and Communication Workshop and Conference, CCWC 2018, 2018-January, pp. 204–208. doi:10.1109/CCWC.2018.8301755.

Koc, L., Mazzuchi, T.A., Sarkani, S., 2012. A network intrusion detection system based on a Hidden Naïve Bayes multiclass classifier. Expert Syst. Appl. 39 (18), 13492–13500. doi:10.1016/j.eswa.2012.07.009.

Kuang, F., Xu, W., Zhang, S., 2014. A novel hybrid KPCA and SVM with ga model for intrusion detection. Appl. Soft Comput. J. 18, 178–184. doi:10.1016/j.asoc.2014.01.028.

Kushwah, G.S., Ali, S.T., 2018. Detecting DDoS attacks in cloud computing using ann and black hole optimization. In: Proceedings of the Second International Conference on Telecommunication and Networks, TEL-NET 2017, 2018-January, pp. 1–5. doi:10.1109/TEL-NET.2017.8343555.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H., 2017. Feature selection: a data perspective. ACM Comput. Surveys 50 (6). doi:10.1145/3136625.

Li, J., Liu, Y., Gu, L., 2010. Ddos attack detection based on neural network. In: Proceedings of the Second International Symposium on Aware Computing, ISAC 2010 - Symposium Guide, pp. 196–199. doi:10.1109/ISAC.2010.5670479.

Li, L., Yu, Y., Bai, S., Hou, Y., Chen, X., 2017a. An effective two-step intrusion detection approach based on binary classification and k-NN. IEEE Access 6, 12060–12073. doi:10.1109/ACCESS.2017.2787719.

Li, Y., Li, T., Liu, H., 2017b. Recent advances in feature selection and its applications. Knowl. Inf. Syst. 53 (3), 551–577. doi:10.1007/s10115-017-1059-8.

Li, Y., Xia, J., Zhang, S., Yan, J., Ai, X., Dai, K., 2012. An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst. Appl. 39 (1), 424–430. doi:10.1016/j.eswa.2011.07.032.

Lin, S.W., Ying, K.C., Lee, C.Y., Lee, Z.J., 2012. An intelligent algorithm with feature selection and decision rules applied to anomaly intrusion detection. Appl. Soft Comput. J. 12 (10), 3285–3290. doi:10.1016/j.asoc.2012.05.004.

Lu, Y., Wang, M., 2016. An easy defense mechanism against botnet-based DDoS flooding attack originated in SDN environment using sFlow. In: Proceedings of the ACM International Conference Proceeding Series, 15-17-June-2016, pp. 14–20. doi:10.1145/2935663.2935674.

Mahbod, T., Ebrahim, B., Wei, L., Ali, A.G., 2009. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, pp. 1–6. doi:10.1109/CISDA.2009.5356528.

Manavi, M.T., 2018. A survey of defense mechanisms against distributed denial of service(DDoS) flooding attacks. Comput. Electr. Eng. 72 (4), 26–38. doi:10.1016/j.compeleceng.2018.09.001.

Monirul Kabir, M., Monirul Islam, M., Murase, K., 2010. A new wrapper feature selection approach using neural network. Neurocomputing 73 (16–18), 3273–3283. doi:10.1016/j.neucom.2010.04.003.

Nielsen, M.A., 2015. Improving the way neural networks learn. Neural Networks and Deep Learning. Determination Press.

Osanaiye, O., Cai, H., Choo, K.K.R., Dehghantanha, A., Xu, Z., Dlodlo, M., 2016. Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. EURASIP J. Wirel. Commun. Netw. 2016 (1). doi:10.1186/s13638-016-0623-3.

Perakovic, D., Perisa, M., Cvitic, I., Husnjak, S., 2017. Artificial neuron network implementation in detection and classification of DDoS traffic. In: Proceedings of the Twenty-Fourth Telecommunications Forum, TELFOR 2016 doi:10.1109/TELFOR.2016.7818791.

Romero, E., Sopena, J.M., 2008. Performing feature selection with multilayer perceptrons 19 (3), 431–441. doi:10.1109/TNN.2007.909535.

Saad, R.M.A., Anbar, M., Manickam, S., Alomari, E., 2016. An intelligent ICMPv6 DDoS flooding-attack detection framework (v6IIDS) using back-propagation neural network. IETE Tech. Rev. 33 (3), 244–255. doi:10.1080/02564602.2015.1098576.

Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., Hakimian, P., 2011. Detecting p2p botnets through network behavior analysis and machine learning. In: Proceedings of the Nineth Annual International Conference on Privacy, Security and Trust, PST 2011, pp. 174–180. doi:10.1109/PST.2011.5971980.

Sabar, N.R., Yi, X., Song, A., 2018. A bi-objective hyper-heuristic support vector machines for big data cyber-security. IEEE Access 6, 10421–10431. doi:10.1109/ACCESS.2018.2801792.

Saied, A., Overill, R.E., Radzik, T., 2016. Detection of known and unknown DDoS attacks using artificial neural networks. Neurocomputing 172, 385–393. doi:10.1016/j.neucom.2015.04.101.

Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. 31 (3), 357–374. doi:10.1016/j.cose.2011.12.012.

Silva, S., 2013. Botnets: a survey. Comput. Netw. 57 (2), 378–403. doi:10.1016/j.comnet.2012.07.021.

Sindhu, S., S., S., Geetha, S., Kannan, A., 2012. Decision tree based light weight intrusion detection using a wrapper approach. Expert Syst. Appl. 39 (1), 129–141. doi:10.1016/j.eswa.2011.06.013.

Singh, K., Singh, P., Kumar, K., 2017. Application layer http-get flood DDoS attacks: research landscape and challenges. Comput. Secur. 65, 344–372. doi:10.1016/j.cose.2016.10.005.

Singh, K.J., Thongam, K., De, T., 2016. Entropy-based application layer DDoS attack detection using artificial neural networks. Entropy 18 (10). doi:10.3390/e18100350.

Singh, R., Kumar, H., Singla, R.K., 2015. An intrusion detection system using network traffic profiling and online sequential extreme learning machine. Expert Syst. Appl. 42 (22), 8609–8624. doi:10.1016/j.eswa.2015.07.015.

Stevanovic, D., Vlajic, N., An, A., 2013. Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Appl. Soft Comput. J. 13 (1), 698–708. doi:10.1016/j.asoc.2012.08.028.

Suresh, M., Anitha, R., 2011. Evaluating machine learning algorithms for detecting DDoS attacks. Comm. Com. Inf. Sc. 196 CCIS, 441–452.

Tang, P.S., Tang, X.L., Tao, Z.Y., Li, J.P., 2014. Research on feature selection algorithm based on mutual information and genetic algorithm. In: Proceedings of the Eleventh International Computer Conference on Wavelet Active Media Technology and Information Processing, ICCWAMTIP 2014, 1, pp. 403–406. doi:10.1109/ICCWAMTIP.2014.7073436.

Vesa, O., Marius, T., Jukka, S., 2001. Feature selection method using neural network. In: Proceedings of the International Conference on Image Processing, pp. 513–516.

Wang, C., Zheng, J., Li, X., 2017. Research on DDoS attacks detection based on RDF-SVM. In: Proceedings of the Tenth International Conference on Intelligent Computation Technology and Automation, ICICTA 2017, 2017-October, pp. 161–165. doi:10.1109/ICICTA.2017.43.

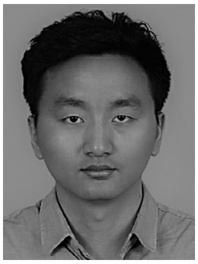
Windeatt, T., Duangsoithong, R., Smith, R., 2011. Embedded feature ranking for ensemble MLP classifiers. IEEE Trans. Neural Netw. 22 (6), 988–994. doi:10.1109/TNN.2011.2138158.

Yusof, A.R.A., Udzir, N.I., Selamat, A., Hamdan, H., Abdullah, M.T., 2018. Adaptive feature selection for denial of services (dos) attack. In: Proceedings of the IEEE

Conference on Applications, Information and Network Security, AINS 2017, 2018-January, pp. 1–4. doi:10.1109/AINS.2017.8270429.

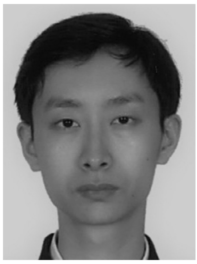
Zhang, F., Wang, D., 2013. An effective feature selection approach for network intrusion detection. In: Proceedings of the IEEE Eighth International Conference on Networking, Architecture and Storage, NAS 2013, pp. 307–311. doi:10.1109/NAS.2013.49.

Zhao, T., Lo, D.C.T., Qian, K., 2015. A neural-network based DDoS detection system using hadoop and hbase. In: Proceedings of the IEEE Seventeenth International Conference on High Performance Computing and Communications, pp. 1326–1331. doi:10.1109/HPCC-CSS-ICESS.2015.38.



Meng Wang received a bachelors degree in electronic and information engineering from HuBei University of Technology, Wuhan, China, in 2012. He is currently pursuing the Ph.D. degree in the School of Electronics and Information Engineering, South China University of Technology, Guangzhou, China. His current research interests are in network security and Software-Defined Networking.

Yiqin Lu was born in ZhaoQin, Guangdong, China. He received a Ph.D. degree in electronic circuits and systems form South China University of Technology (SCUT), Guangzhou, China, in 1996. His research interests include communication network, network security, error-correcting code, and Internet of things. He was appointed a professor of SCUT in 2006. He is also the doctoral supervisor of the college of electronic and information in SCUT in Guangzhou, China. He is now the dean of the office of the Leading Group for Cyberspace Affairs of SCUT, the director of the Network Center of Southern China Region of China Education and Research Network (CERNET), and the president of Computer Information Network Safety Association of Guangdong Province.



Jiancheng Qin received the bachelor's degree in computer engineering from the School of Computer Science and Technology, Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1999, and the master's degree from the School of Software Engineering, BUPT, in 2008, and the Ph.D. degree from School of Computer, BUPT, in 2011. His major fields of study are information security and mobile computing.