IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received December 29, 2019, accepted January 12, 2020, date of publication January 17, 2020, date of current version January 28, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2967478

Low-Rate DDoS Attack Detection Based on Factorization Machine in Software Defined Network

WU ZHIJUN, XU QING, WANG JINGJIE, YUE MENG, AND LIU LIANG
School of the Electronic Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Wu Zhijun (caucwu@263.net)

This work was supported in part by the Joint Foundation of National Natural Science Committee of China and Civil Aviation Administration of China under Grant U1933108, in part by the Key Program of Natural Science Foundation of Tianjin under Grant 17JCZDJC30900, in part by the National Science Foundation for Young Scientists of China under Grant 61601467, in part by the Scientific Research Project of Tianjin Municipal Education Commission under Grant 2019KJ117, and in part by the Fundamental Research Funds for the Central Universities of China under Grant 3122018D007 and Grant 3122018C003.

ABSTRACT As the Software Define Network (SDN) adopts centralized control logic, it is vulnerable to various types of Distributed Denial of Service (DDoS) attacks. At present, almost all the research work focuses on high-rate DDoS attack against the SDN control layer. Moreover, most of the existing detection methods are effective for high-rate DDoS attack detection of the control layer, while a low-rate DDoS attack against the SDN data layer is highly concealed, and the detection accuracy against this kind of attack is low. In order to improve the detection accuracy of the low-rate DDoS attack against the SDN data layer, this paper studies the mechanism of such attacks, and then proposes a multi-feature DDoS attack detection method based on Factorization Machine (FM). The features extracted from the flow rules are used to detect low-rate DDoS attacks, and the detection of low-rate DDoS attacks based on FM machine learning algorithms is implemented. The experimental results show that the method can effectively detect the low-rate DDoS attack against the SDN data layer, and the detection accuracy reaches 95.80 percent. Because FM algorithm can achieve fine-grained detection for low-rate DDoS attack, which provides a reliable condition for defending against such attacks. Finally, this paper proposes a defense method based on dynamic deletion of flow rules, and carries out experimental simulation and analysis to prove the effectiveness of the defense method, and the success rate of forwarding normal packets reached 97.85 percent.

INDEX TERMS Low-rate denial of service, factorization machine, software defined network, detection, multi-feature.

I. INTRODUCTION

In recent years, big data and cloud computing and other network technologies have achieved rapid development. SDN (Software Defined network) has attracted the interest of researchers because of its new network paradigm [1], and it has also been greatly developed in many companies, such as Tencent and Alibaba's cloud computing platform has been widely used. Since the control plane and the data plane of the SDN are in a separated state, the centralized control of the controller and the distribution and forwarding of the switch are realized, which greatly simplifies the management of the network and improves the innovation capability of the network, and has greater advantages than the traditional network. However, since SDN is divided into application layer, control layer and data layer, the possibility of SDN being attacked is also increased. On the one hand, the functions of SDN (such as centralized control, global view) make it easier to detect and respond quickly to DoS attacks. On the other hand, the vulnerability of data plane also provides favorable conditions for new DoS attacks. Therefore, SDN itself may also be the target of DoS attacks. In fact, the entire SDN platform has potential DoS attack vulnerabilities [2]. For example, attackers can use the characteristics of SDN to launch DoS attacks on the control layer. Of course, the application layer and the data layer [3] are also likely to be attacked.

The associate editor coordinating the review of this manuscript and approving it for publication was Fan Zhang.

Therefore, research on DoS attacks detection and defense for SDN networks is becoming more and more important in the field of network security.

Although SDN can provide rich network functions, the network usage efficiency is improved. However, SDN still faces many security challenges [4], [5] at the same time, such as DoS attack, network blocking, switch data leakage, data management confidentiality and other common attacks in traditional networks [6]. The data plane is the traffic entry of the SDN networks, and of course, it is also the traffic entry of DDoS attacks. Compared with DDoS attacks at various layers of SDN, low-rate DDoS attacks initiated on the data plane are characterized by low speed, concealment, and persistence, which makes it difficult to detect. The first problem in the existing solution is that the feature selection is not obvious. Since most of the existing detection schemes are related to the DDoS attack of the SDN control layer, the features are all sampled by using the entire flow table as a sample. In this paper, the characteristics of flow rules are directly targeted, and each flow rule is taken as a sample for sampling. This is because the low-rate DDoS attack is hidden in the normal data flow, and the corresponding flow rule is also hidden in the legal flow rule. The detection with each flow rule as the sample can detect the specific flow rules as the attack flow rules, which improves the fine granularity of detection and achieves the accurate detection effect. The second problem is that the existing solution has low detection accuracy. Therefore, this paper introduces feature combination mechanism by using FM algorithm, the feature combination detection method can establish the correlation between each feature sample, so that the features used to update the parameters are used. The sample is more abundant, which increases the detection rate. In order to verify the advantages of the detection method in this paper, we compare the detection method proposed in this paper with the two detection methods of the existing low-rate DDoS attacks, and prove the reliability of the detection method.

For the problems of low rate DDoS attacks detection accuracy of SDN networks is low and the features are not obvious. This paper first uses the idle timeout mechanism to explain how to launch such an attack and verify its effectiveness. Then, aiming at the problem that the characteristics of low-rate DDoS attack are not obvious, this paper mines four effective characteristics and systematically analyzes them, and proposes an effective detection method for this kind of attack, which improves the detection accuracy. Finally, based on the completion of detection, we proposed a defense method based on dynamic deletion of flow rules, which not only prevented the saturation of flow table, but also improved the forwarding success rate of normal clients and ensured the normal operation of SDN.

II. RELATED WORKS

The DoS attack detection methods for SDN networks at present mainly divided into two categories. One is based on the threshold detection method, which are based on detecting one or several traffic indicators, such as traffic rate, maximum entropy and packet delay. The traffic metrics method detects these metrics in real time, and once the indicator exceeds a predetermined threshold, an attack may occur in the network. Dhawan *et al.* [7] proposed a DoS attack detection method by monitoring the installation rate of flow table rules. Once the installation rate of flow rules exceeds a certain threshold, it indicates that the network may be attacked, and then triggers the defense mechanism. S. m. Mousavi and Sthilaire [8] proposed an early attack detection method based on the entropy value of the destination address. If the entropy is lower than the preset threshold, the algorithm determines that the attack is in progress. He *et al.* [9] proposed an SDCC scheme based on confidence filtering combined with link bandwidth and data flow detection. The scheme calculates the data-group CBF (confidence-based filtering) score and judges the packet below the threshold as attack packets, which also need to establish the SDCC's signature database of attack flow, and maintain the update of the profile table. The process is more complicated. At the same time, in order to reduce resource consumption, the ratio of the sampling of the data flow in the normal, early warning, and defense states of the scheme are 20%, 40%, 80%, however, there is still a risk of false detection and false positives in this method. Liu [10] proposed a detection method of low-rate DDoS in SDN in a cloud environment. This method gradually records the survival time of each flow rule in the flow table by setting a flow table snapshot and a suspicious table in multiple detection periods, and sets alarm threshold, to judge whether it is attacked by low-rate DDoS. In order to prevent the overflow of the flow table, Zhu *et al.* [11] proposed a dynamic timeout mechanism, which dynamically adjusts the timeout through the current state of the flow table. Experimental results show that this mechanism performs well in reducing TCAM occupation. Wang *et al.* [12] proposed a safe-guard scheme (SGS), which deployed multiple controllers on the control plane through clustering algorithm to improve the defense capability of the control plane. However, this scheme required high performance of the switch. A statistical model of the software-defined network score (SDNScore) was proposed in [13]. However, the concept of a capable switch is a controversial issue in the literature, because it conflicts with the centralized control and distributed forwarding characteristics of SDN [14]. Xu *et al.* [15] proposed a DDoS attack defense strategy based on traffic classification (DDTC). The DDTC has three modules, the first module is an attack trigger module, the second module is an attack detection module, and the third module is an attack traceback module. However, the algorithm involved in the attack detection module has a high time complexity, so it needs to be further improved. In general, based on the threshold detection has the advantage of simple implementation. In addition, it does not require complex algorithms to process data, so it performs well in real time. However, such detection typically relies on only a few metrics, so it is easy to mistake a normal random burst in a real network for an attack. In addition, the results of these methods are sensitive to the selection of detection threshold, which needs to be changed according to the network scene, or it will seriously affect the accurate detection probability.

Another type of method is feature-based detection. The essence of this method is to build a classifier to classify normal and attack streams. Typically, statistical analysis and support vector machines are used to process attack signatures and further construct detection models. Braga *et al.* [16] proposed a lightweight DDoS attack detection method based on the traffic 6-tuple feature. They use Self-organizing Maps (SOM) to classify network traffic as normal and abnormal. Experimental results show that the method has a very good detection rate. Wu *et al.* [17] proposed a low-rate DDoS detection method based on joint features, and proposed features such as small packet ratio and packet loss rate, and trained by BP (back propagation) neural network to detect low-rate DDoS attacks. However, when the attack strength is weak, the change of the packet loss rate of the attack traffic is not particularly obvious, which may result in deviation of the detection result. Bu *et al.* [18] used a combination of support vector machine (SVM) and self-organizing mapping (SOM) to detect DDoS attack flows, which improved the accuracy of detecting DDoS attacks. However, the combination of machine learning algorithms greatly increases the computational complexity of the system. Barki *et al.* [19] used different machine learning algorithms to classify incoming requests to detect DDoS attacks, used more accurate machine learning algorithms to implement IDS in SDN networks, and achieved good results. In [20], they also use support vector machine (SVM) classifiers and neural network (NN) classifiers to detect suspicious and harmful connections. Experimental results show that the support vector machine algorithm is a better choice for implementing IDS in SDN networks. Garg *et al.* [21] proposed a hybrid anomaly detection system based on deep learning. This system combines RBM and SVM to reduce dimensions, and then classifies regular and irregular traffic in SDN accordingly. However, the dataset used is KDD'99, not a flow-based dataset. Haider *et al.* [22] proposed a deep learning-based convolutional neural network integration scheme for the detection of DDoS attacks in SDN. The experimental results show that the framework has higher attack detection accuracy, but the scheme has higher computational complexity. Dong *et al.* [23] proposed a secure cluster management architecture based on big data analysis, and proposed a security authentication scheme for cluster management. In addition, they also proposed an ant colony optimization method to make the implementation system of big data analysis scheme and optimization control plane possible. This work is significant in improving the security and efficiency of SDN networks. Feature-based detection methods have higher detection rates and lower error detection rates than threshold-based detection methods.

In summary, most of the current detection methods are used to detect DDoS attacks against the control layer, and there are few detection methods for detecting DDoS attacks against the data layer. In addition, the above several detection methods are excessively dependent on the information provided by the PACKET_IN message. However, when the attacker initiates a low-rate DDoS attack against the SDN data layer, the switch sends fewer PACKET_IN messages to the controller, so it is difficult to achieve the detection effect. The features extracted for the characteristics of the attack flow cannot accurately reflect the characteristics of the attack and possible changes. This paper adopts the detection method based on FM machine learning to extract the four-dimensional features according to the attack characteristics and improve the detection accuracy according to the correlation between the features.

III. DETECTION OF LOW-RATE DDoS ATTACKS IN SDN NETWORKS

This section is divided into three parts. The first part is a low-rate DDoS attack against the data layer. We show how to launch this attack. The second part details the extraction of low-rate DDoS attack features in SDN. The third part shows how to apply the extracted features to FM algorithm to detect low-rate DDoS attacks. The fourth section explains in detail how to defend against such low-rate DDoS attacks.

A. LOW-RATE DDoS ATTACKS AGAINST THE DATA LAYER

DDoS attacks against the control require very high rates, while denial-of-service attacks can also create a covert attack through low-rate traffic. In fact, there is a low-rate DDoS attack against the application layer in the traditional network, such as the attacker denies the service by sending a very low-rate request to a VoIP server [24]. Attackers can also use SlowDroid [25] to initiate low-rate attacks on devices that are not very powerful, and exploit vulnerabilities on the application layer protocol to evade existing detection defense mechanisms, such as SlowNext [26].

In SDN, the controller is a powerful component. It is responsible for formulating forwarding rules for packets in the entire network. It has powerful processing capabilities, so low-rate DDoS attacks cannot pose a threat. However, the data layer of SDN mainly refers to the OpenFlow switch, which is only responsible for the forwarding of data packets and the storage of flow table rules, and its processing capability is limited. In addition, flow table rules are typically stored in a Ternary Content-Addressable Memory (TCAM). This type of memory is expensive [27], and OpenFlow switches have very limited TCAM and can only store 1500-3000 flow rules [28]. Therefore, the data layer of SDN is easy to become the target of low-rate DDoS attacks.

Inspired by the low-rate application layer DDoS attack and the characteristics of the SDN data layer, T A. Pascoal et al. [29] proposed a low-rate DoS attack for the SDN data layer, called Slow TCAM Exhaustion attack, which is the low-rate DDoS attack for the data layer studied in this paper. The attack needs to meet the following four conditions.

i. The attacker needs to control a certain number of bots, the number should be a little more than half of the capacity

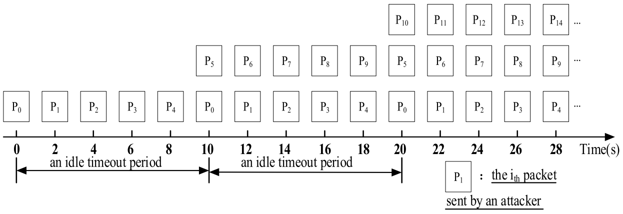


FIGURE 1. DDoS attacks against the data layer.

of switch's rules, and the attacker will not send packets with spoofed IPs.

ii. Each bot needs to send the same packet with the real address to the attacked switch. When the switch receives a new packet, the corresponding flow rule will be installed in the switch. Eventually there will be two flow rules in the switch.

iii. The rate at which each packet is sent by the bot is not too high. The attacker generates only a few dozen packets per second, the DDoS attack against the controller generates more than 1000 packets per second, and these packets have fake IP addresses.

iv. The interval at which each bot sends a data packet should be less than the idle timeout of the flow entry. The idle timeout period of a flow entry is the maximum time that the flow entry can be in the no-match state. If the flow rule does not have a matching packet during this time, the flow rule is automatically cleared. Therefore, as long as the transmission rate is large and the reciprocal of the idle timeout period is guaranteed, the existence of the flow entry and the switch can be guaranteed. Eventually, when the switch's flow table capacity is full, it cannot install new flow rules and affect the forwarding of new packets.

The low-rate DDoS attacks studied in this paper use a linear incremental increase in the number of low-rate attack flows, as shown in Fig.1. Assume that the attacker sends 5 attack packets in the first idle timeout interval, and at the same time, 10 flow rules will be generated in the flow table. Then in the second idle timeout interval, in order to ensure that the flow rules in the flow table do not disappear, the attack packet is sent periodically, but 5 attack packets are added at the same time. At this time, there will be 20 flow rules in the flow table. As the attack continues, the flow rules will gradually increase until the flow table is saturated.

B. FEATURE EXTRACTION OF LOW-RATE DDoS ATTACKS IN SDN

Low-rate DDoS attacks for the data layer are quite different from high-rate DDoS attacks for the control layer. Low-rate DDoS attacks only launch attacks at a lower rate by controlling a smaller number of bots. Although the attack target is not the control layer, it will have a certain degree of impact on the switches in the data layer. This kind of attack will be more covert. It is not easy to find obvious features in the flow table, and it is less easy to be detected. Since most of the existing detection schemes are related to the DDoS attack of the SDN control layer, the features are all sampled by using the entire flow table as a sample. In this paper, the characteristics of flow rules are directly targeted, and each flow rule is taken as a sample for sampling. This is because the low-rate DDoS attack is hidden in the normal data flow, and the corresponding flow rule is also hidden in the legal flow rule. The detection with each flow rule as the sample can detect the specific flow rules as the attack flow rules, which improves the fine granularity of detection and achieves the accurate detection effect.

For the four eigenvalues that do not need to be processed, including duration time, packets number, relative dispersion of match bytes, and relative dispersion of packet interval, the meanings of the eigenvalues are as follows.

i. Duration time: The duration time of a flow rule refers to the time from the appearance of the flow rule to the current flow table. The legal flow rule usually has a short time in the flow table and will not exist in the switch for a long time. Literature [30] pointed out that the duration time of 0.1% flow rules in the data center can reach 200 s, while the duration time of 80% flow rules is about 10 s. The purpose of the low-rate DDoS attack on the data layer in SDN is to exhaust the flow table resources, so this attack will send attack packets all the time to occupy the flow table for a long time. Therefore, we take the duration time of the flow rule as one of the detection features.

ii. Packets number: The total number of packets matched by the flow rule refers to how many packets the flow rule has matched in the duration time. In the low-rate DDoS attack against the data layer, since the attack flow rule needs to be continuously matched, the field value of the attack flow rule is larger than the value of the field of the legal flow rule. Therefore, we use the packets number as the detection feature.

iii. Relative dispersion of match bytes (RDMB): Another difference between attack flow rule and normal flow rule is that the contents of the packet. In order to meet the normal access requirements of users, packets with normal flow usually match large bytes with large variance, while the attack traffic is just to occupy the flow table resources and has no practical significance, so the bytes of packets are fewer and little changed. Hence, we propose relative dispersion of match bytes as one of the characteristics. The calculation formula of relative dispersion of match bytes is shown below.

$$RDMB = \frac{\sum_{i=1}^{N}(X_i - \mu)^2}{N} \tag{1}$$

where, N represents the number of packets matched by the flow rules in the window function, $X_i\,\{i = 1 \ldots N\}$ represents the size of bytes matched by each packet, μ represents the mean value of matched bytes sizes within the window function.

iv. Relative dispersion of packet interval (RDPI): From the DoS attack model proposed in the paper, we can see that the packet delivery interval of the DoS attack traffic is periodic and slightly smaller than timeout. As for the normal user, due to the objective needs, they will send multiple data packets per unit time, and the interval of these data packets is random and different. So we take the relative dispersion of packet interval as the detection feature. The calculation formula of relative dispersion of packet interval is formula (2).

$$RDPI = \frac{\sum_{i=1}^{M} |T_i - \lambda|}{M} \tag{2}$$

where, M is the number of packet interval in the window function, T_i {i = 1 . . . M} represent the time of each packet interval, λ means the mean of packet interval in the window function.

The reason is to sample each flow rule as a sample, rather than the whole flow table as a sample for statistical sampling, as in the detection of DDoS attack on the controller. This is because low-rate DDoS attacks are hidden in normal data flows, and the corresponding flow rules are also hidden in legitimate flow rules. Taking each flow rule as a sample for detection, specific flow rules can be detected as attack flow rules, to determine the attack flow rules and achieve accurate detection effect.

C. DOS ATTACK DETECTION ALGORITHM BASED ON FM

In 2010, Steffen Rendle first proposed the FM (Factorization Machine) [31] algorithm, mainly to solve the feature combination problem under sparse data. With its large amount of data and sparse features, it can still get excellent performance and effective characteristics. However, this does not mean that FM is only suitable for sparse data processing. FM is a more generalized and general model. It can be applied to many prediction tasks, such as regression, classification, ranking and so on. In this paper, the feature combination mechanism is introduced through FM algorithm to establish the correlation between each feature sample, so that the actual feature samples used to update parameters are more abundant, thereby improving the detection rate and real-time detection of attack flow rules, thus providing reliable conditions for resisting low-rate DDoS attacks against the SDN data layer.

Assume that the extracted flow rule feature sample is $X = \{x_1, x_2, \ldots, x_4\}$: x_1 to x_4 represent duration time, packets number, relative dispersion of match bytes, relative dispersion of packet interval. The expression of the general linear model prediction flow rule is formula (3).

$$y = w_0 + \sum_{i=1}^{n} w_i x_i \tag{3}$$

where, n is the feature dimension, $n = 4$ in this paper, x_i is the input feature sample, w_i is the modified weight, and w_0 is the initial weight.

In the general linear model, each feature is considered independently, and the relationship between features and features is not considered. But, when the network is under attack, there is a correlation between many features. For example, in the low-rate DDoS attack flow rule feature for the data layer, the number of matching packets and the number of matching bytes is often related. In general, the more matching the number of matching bytes, the more matching bytes. In order to combine the flow rule features, a polynomial model needs to be introduced, such as equation (4).

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} w_{ij} x_i x_j \tag{4}$$

where, w_{ij} is a combined feature parameter. Since w_{ij} cannot be obtained through training, and the corresponding parameters cannot be carried out. Therefore, the method used here is to introduce an auxiliary vector $v_i = (v_{i1}, v_{i2}, v_{i3}, \ldots, v_{ik})$ for each feature vector x_i. Then use $v_i v_j^T$ to estimate the parameter of the cross term coefficient w_{ij}, that is, $\hat{w} = v_i v_j^T$, the formula (5) is obtained.

$$y = w_0 + \sum_{i=1}^{n} w_i x_i + \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} < v_i, \quad v_j > x_i x_j \tag{5}$$

where $v \in R^{n,k}$, v_i is the implicit vector of the i^{th} dimensional feature, representing the dot product between two vectors of size k, and its calculation formula is as in formula (6).

$$< v_i, v_j > = \sum_{f=1}^{k} v_{i,f} \bullet v_{j,f} \tag{6}$$

where, $v_{i,f}$ is the element of the f^{th} of the hidden vector v_i, and $v_{j,f}$ is the element of the f^{th} of the hidden vector v_j. Compared with the linear model, the FM has more features combined with the latter. This part can produce more combined features, which provide a solid foundation for subsequent more accurate classification. All $v_{j,f}$ can form a matrix V. The expression of V is as shown in equation (7). It is a matrix of n rows and k columns, and each row represents a hidden vector $\vec{v}_i$.

$$V = \begin{pmatrix} v_{11} & v_{12} & \ldots & v_{1k} \\ v_{21} & v_{22} & \ldots & v_{2k} \\ \vdots & \vdots & & \vdots \\ v_{n1} & v_{n2} & \ldots & v_{nk} \end{pmatrix} = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_n \end{pmatrix} \tag{7}$$

then,

$$W = VV^r = \begin{pmatrix} \vec{v}_1 \\ \vec{v}_2 \\ \vdots \\ \vec{v}_n \end{pmatrix} \begin{pmatrix} \vec{v}_1^T & \vec{v}_2^T & \ldots & \vec{v}_n^T \end{pmatrix} \tag{8}$$

As shown in equation (8), it is equivalent to matrix decomposition of W, where $\vec{V}_i$ is an implicit vector and k represents the length of the hidden vector. The definition of k value has a certain influence on the classification accuracy of FM. The selection principle of k value has been described in detail in [31]. Since the feature dimension should be less than an order of magnitude of the training data, our training data is 140,000, and the feature dimension should be around 10,000. Since there are more than 500 eigenvalues, we set the k values to 20 and 30, which ensures that the training is adequate.

Algorithm 1 SGD

```
Input: Training data set $S$, regularization parameters $\lambda$, learning rate $\eta$, initialization $\delta$
Output: Model parameters $\Theta = (w_0, \mathbf{w}, \mathbf{V})$
$w_0 \leftarrow 0$; $\mathbf{w} \leftarrow (0, \ldots, 0)$; $\mathbf{V} \sim N(0, \sigma)$;
repeat
  for$(\mathbf{x}, y) \in S$ do
    $w_0 \leftarrow w_0 - \eta \left( \frac{\partial}{\partial w_0} l\left(\hat{y}\left(\mathbf{x}|\Theta\right), y\right) + 2\lambda^0 w_0 \right)$;
    for $i \in \{1, \ldots, p\} \wedge x_i \neq 0$ do
      $w_0 \leftarrow w_0 - \eta \left( \frac{\partial}{\partial w_0} l\left(\hat{y}\left(\mathbf{x}|\Theta\right), y\right) + 2\lambda^w_{\pi(i)} w_i \right)$;
      for $f \in \{1, \ldots, k\}$ do
        $v_{i,f} \leftarrow v_{i,f} - \eta \left( \frac{\partial}{\partial v_{i,f}} l\left(\hat{y}\left(\mathbf{x}|\Theta\right), y\right) + 2\lambda^v_{f,\pi(i)} v_{i,f} \right)$;
      end
    end
  end
until stopping criterion is met;
```

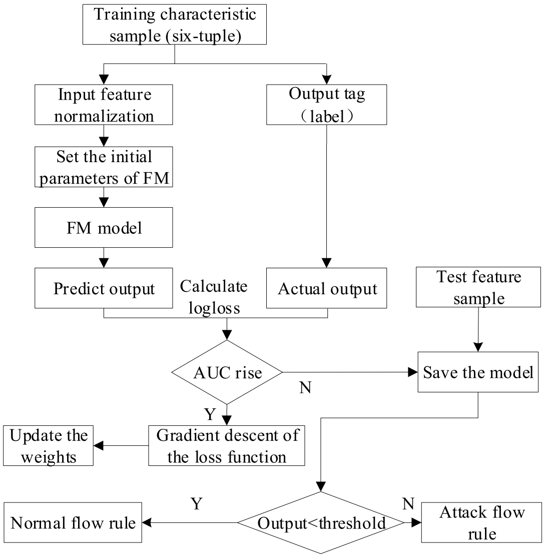


FIGURE 2. DoS attack detection method based on FM algorithm.

For most loss functions, the Stochastic Gradient Descent (SGD) algorithm solves the parameters w and v in FM. Here is the pseudo-code portion of the algorithm.

Because there are many gradient descent methods, the stochastic gradient descent method (SGD) is used in this paper to find the optimal value of ω_0, $\mathbf{w}$, $\mathbf{V}$, and the predicted value is closest to the true value at this time. The process of gradient descent is to move the starting point in the opposite direction of the gradient, and then move the learning rate multiplied by the step value of the gradient until the optimal solution is found. To avoid overfitting, we introduce a regular term in the gradient. For any sample ($\mathbf{x}$, y), first update the gradient of ω_0, then complete the gradient update of $\mathbf{w}$ and $\mathbf{V}$ by loop iteration, and finally get the optimal solution of three parameters.

The gradient calculation formula of the FM two-class problem model is shown in equation (9).

$$\begin{aligned} \frac{\partial loss^c(\hat{y}, y)}{\partial \theta} &= -\frac{1}{\delta(\hat{y}y)} \delta(\hat{y}y) \bullet [1 - \delta(\hat{y}y)] \bullet y \bullet \frac{\partial \hat{y}}{\partial \theta} \\ &= [\delta(\hat{y}y) - 1] \bullet y \bullet \frac{\partial \hat{y}}{\partial \theta} \end{aligned} \tag{9}$$

The calculation formula of $\frac{\partial \hat{y}}{\partial \theta}$ is as shown in formula (10).

$$\frac{\partial \hat{y}}{\partial \theta} = \begin{cases} 1, & \text{if } \theta = w_0 \\ x_i, & \text{if } \theta = w_i \\ x_i \sum_{j=1}^{n} v_{j,f} x_j - v_{i,f} x_i^2, & \text{if } \theta = v_{i,f} \end{cases} \tag{10}$$

where, $\sum_{j=1}^{n} v_{j,f} x_j$ can be pre-set.

The flow of the DoS attack detection method based on the FM algorithm is shown in Fig.2. Each feature sample is a four-tuple and needs to be tagged, normally labeled 0, and the attack tag is 1. When initializing parameters, the parameters that need to be initialized are offset weight w_0, one-item weight w and the hidden vector dimension k of the cross terms. The setting of these parameters can be found in [31]. The loss function of the algorithm uses the Sigmod function, and the gradient descent uses the stochastic gradient descent algorithm.

Since the input sample is sampled in each flow entry, the algorithm can detect the attack flow rules accurately, so that a finer-grained DoS attack detection can be performed. First, we collected 140,000 training data and selected 100,000 and 40,000 sets of features as training and test sets. The test set is tagged according to the source port, ports 45132 to 45698 are attack flow rules, labeled 1, ports 50000 through 50750 are normal flow rules, labeled 0. Next, the feature is hashed to the specified order of magnitude space, and the training parameters, the learning rate, and the dimension k of the hidden vector are set separately. Adjust the parameters and training model, get the best model based on logloss and AUC and save the model. The word model is used to predict the traffic of the test set and to determine if there is an attack based on the predicted probability.

D. A DEFENSE METHOD BASED ON DYNAMIC DELETION OF FLOW RULES

The OpenFlow protocol can support the controller to manage the flow table in the data layer. For example, the target flow rule can be deleted through the FLOW_MOD message. Based on this theoretical basis, this paper proposes a defense method for dynamically deleting flow rules, which is a defense method against DDoS attacks in the data layer. The method is based on the successful detection of low-rate DDoS attacks against the data layer. Only the DoS attack detection algorithm can accurately detect the attack flow rules, and the defense method can be effectively performed. Therefore, both DoS attack detection methods and defense methods need to be deployed in the controller.

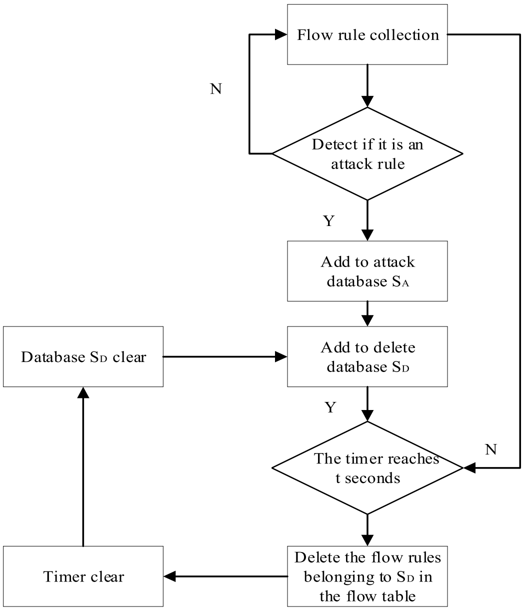

Flow rule collection
N
Detect if it is an attack rule
Y
Add to attack database SA
Database SD clear
Add to delete database SD
Y
N
The timer reaches t seconds
Timer clear
Delete the flow rules belonging to SD in the flow table

FIGURE 3. A defense method based on dynamic deletion of flow rules.

The strategy flow chart of the controller's dynamic deletion of flow rules is shown in Fig.3. Firstly, the sample of flow rules is collected, and then the trained FM algorithm model is put into it to test whether it is attack flow rules. If it is an attack flow rule, add it to the attack database SA and delete the database SD respectively. Then judge whether the timer has reached t seconds (that is, the time interval for deleting the flow rule). If it has reached t seconds, the collection of the flow rule will continue. If it reaches *t* seconds, the FLOW_MOD message will be sent to the switch under the command of deleting all flow rules in SD, so that the number of flow rules in the switch will drop, and then the counter and database SD will be cleared.

IV. EXPERIMENT AND ANALYSIS OF RESULTS

In this paper, we first use Mininet [32] and Ryu [33] to establish an SDN network simulation platform, and under the simulation platform, the server targets the data layer with low-rate DDoS attacks. Experimental verification and analysis of FM-based DoS attack detection methods.

A. EXPERIMENTAL ENVIRONMENT

The experimental network topology is shown in Fig. 4.

In this experiment, two real hosts are used. One of the hosts is responsible for running Mininet. The version used in this experiment is 2.3.2. The Mininet also comes with Open Virtual Switch (OVS). The version of OVS is 2.5.0. Mininet is the standard platform for SDN network emulation, and OVS is an open source virtual machine and supports the OpenFlow protocol. The other host is responsible for running the Ryu controller, version 4.19, which is an interface-rich SDN controller that allows researchers to develop versatile control applications using the Python language. The host system running Mininet is Ubuntu 16.04 LTS, its processor is Intel(R) Xeon(R) E3-1225 v6 @ 3.3GHz, and it has 8G running memory; the host system running Ryu controller is Ubuntu 16.04 LTS, its processing The device is Intel(R) Xeon(R) E3-1225 v6 @ 3.3GHz and has 4G running memory. The southbound interface protocol used in the experiment is the OpenFlow protocol, and its version is OpenFlow1.3.

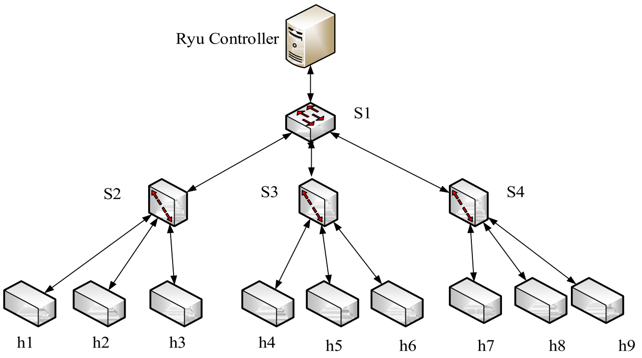


FIGURE 4. Experimental topology.

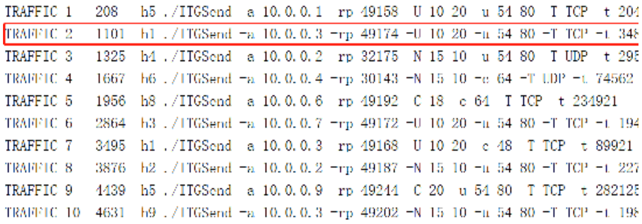
In this experiment, the custom script is created by using Python language in Mininet to establish the experimental network shown in Fig.4. The experimental network topology by using more complex tree network structure, all hosts connected to the switch at the same time set up experiments link bandwidth are 10 Mbps, the network delay is 5 ms, the link loss is 0, the maximum queue size is 1000. The link bandwidth connected to the switch is 50Mbps, the network delay is 10ms, the link loss is 0, and the maximum queue size is 3000.

1) DESIGN OF BACKGROUND TRAFFIC AND ATTACK TRAFFIC

The experiment uses a Distributed Internet Traffic Generator (D-ITG) [34] to generate background traffic, ie normal network traffic. The ITGSend command is executed on each host node to send traffic, and the ITGRecv command is executed to accept traffic. The D-ITG version used in the experiment is D-ITG-2.8.1-r1023.

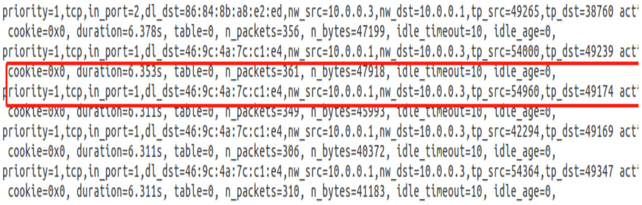
In order to make the background traffic more close to the real traffic, According to our analysis of the CAIDA data set in [35], we inject 100 flows into SDN networks as the background flow of the experimental network, the transmission rate of each flow follows the Poisson distribution, normal distribution, uniform distribution and constant. Among the 100 background flows, TCP flows account for 80%, ICMP flows account for 15% and UDP account for 5%, the average rate of background traffic is about 1.3Mbps.

As shown in Fig.5, there are ten of the background flows generated by the D-ITG. Each row in the figure represents the background traffic in an experimental network. Taking the red box as an example, the second background flow is followed by the third background flow. The time of injection into the network is 1101 milliseconds, and the destination address is 10.0.0.3, which is h3. The destination port is 49174,



```
TRAFFIC 1   208   h5 ./ITGSend  a 10.0.0.1  rp 49158  U 10 20  u 54 80  T TCP  t 204
TRAFFIC 2   1101  h1 ./ITGSend -a 10.0.0.3 -rp 49174 -U 10 20 -u 54 80 -T TCP -t 348
TRAFFIC 3   1325  h4 ./ITGSend  a 10.0.0.2  rp 32175  N 15 10  u 54 80  T UDP  t 295
TRAFFIC 4   1667  h6 ./ITGSend -a 10.0.0.4 -rp 30143 -N 15 10 -c 64 -T UDP -t 74562
TRAFFIC 5   1956  h8 ./ITGSend  a 10.0.0.6  rp 49192  C 18  c 64  T TCP  t 234921
TRAFFIC 6   2864  h3 ./ITGSend -a 10.0.0.7 -rp 49172 -U 10 20 -u 54 80 -T TCP -t 194
TRAFFIC 7   3495  h1 ./ITGSend  a 10.0.0.3  rp 49168  U 10 20  c 48  T TCP  t 89921
TRAFFIC 8   3876  h2 ./ITGSend -a 10.0.0.2 -rp 49187 -N 15 10 -u 54 80 -T TCP -t 227
TRAFFIC 9   4439  h5 ./ITGSend  a 10.0.0.9  rp 49244  C 20  u 54 80  T TCP  t 282125
TRAFFIC 10  4631  h9 ./ITGSend -a 10.0.0.3 -rp 49202 -N 15 10 -u 54 80 -T TCP -t 198
```

FIGURE 5. D-ITG background traffic (partial screenshot).



```
priority=1,tcp,in_port=2,dl_dst=86:84:8b:a8:e2:ed,nw_src=10.0.0.3,nw_dst=10.0.0.1,tp_src=49265,tp_dst=38760 act
 cookie=0x0, duration=6.378s, table=0, n_packets=356, n_bytes=47199, idle_timeout=10, idle_age=0,
priority=1,tcp,in_port=1,dl_dst=46:9c:4a:7c:c1:e4,nw_src=10.0.0.1,nw_dst=10.0.0.3,tp_src=54000,tp_dst=49239 act
 cookie=0x0, duration=6.353s, table=0, n_packets=361, n_bytes=47918, idle_timeout=10, idle_age=0,
priority=1,tcp,in_port=1,dl_dst=46:9c:4a:7c:c1:e4,nw_src=10.0.0.1,nw_dst=10.0.0.3,tp_src=54960,tp_dst=49174 act
 cookie=0x0, duration=6.311s, table=0, n_packets=349, n_bytes=45993, idle_timeout=10, idle_age=0,
priority=1,tcp,in_port=1,dl_dst=46:9c:4a:7c:c1:e4,nw_src=10.0.0.1,nw_dst=10.0.0.3,tp_src=42294,tp_dst=49169 act
 cookie=0x0, duration=6.311s, table=0, n_packets=306, n_bytes=40372, idle_timeout=10, idle_age=0,
priority=1,tcp,in_port=1,dl_dst=46:9c:4a:7c:c1:e4,nw_src=10.0.0.1,nw_dst=10.0.0.3,tp_src=54364,tp_dst=49347 act
 cookie=0x0, duration=6.311s, table=0, n_packets=310, n_bytes=41183, idle_timeout=10, idle_age=0,
```

FIGURE 6. Flow table in the switch (partial screenshot).

the number of packets sent is uniformly distributed from 10 to 20. The size of the transmitted packet is from 54 to 80. The traffic protocol is TCP, and the duration of the traffic is 348247 milliseconds.

As shown in Fig.6, when the background traffic is running, a partial screenshot of the flow table in the OpenFlow switch can be seen from the red box. The flow entry is h1 (IP address is 10.0.0.1) to h3 (IP address is 10.0.0.3). And the destination port is a flow entry of 49174, which corresponds to Fig.5, which can indicate that the background traffic has been successfully injected into the SDN networks.

Scapy [36] is a feature-rich distribution tool and supports Python script programming. It can forge arbitrary fields of packet messages and control the rate of packet transmission. Scapy is used to write attack scripts to achieve low-rate DDoS attacks against the data layer. The average attack rate is 1.6Mbps, 1.9Mbps and 2.2Mbps.

B. EXPERIMENTS AND RESULTS ANALYSIS

The FM-based DoS attack detection method is used to detect low-rate DDoS attacks against the data layer. This section first analyzes the attack characteristics of low-rate DDoS attacks, and analyzes the difference between this attack and high-rate DDoS attacks, and the impact on the SDN networks. Then an index for judging machine learning performance is proposed, and the effectiveness of the FM-based DoS attack detection method is verified by comparing other machine learning algorithms with other low-rate attack detection methods. Finally, based on this, a defense method based on dynamic deletion of flow rules is proposed and the effectiveness of the method is verified.

1) EXPERIMENTAL RESULTS OF LOW-RATE DDoS ATTACKS IN SDN

A high-rate DDoS attack and a low-rate DDoS attack are respectively launched on h1. The average attack rate is 10 Mbps and 1.6 Mbps, and the attack target is h9. The upper limit of the flow rule storage limit of the s1 switch is set to 1500. Then, the number of flow rules in the s1 switch under the normal network and the two attack networks is separately calculated, and Fig.7 is obtained. At the same time, the number of FLOW_MOD messages sent by the controller to the s1 switch in the normal network and the two attack networks is counted, and Fig.7 is obtained.

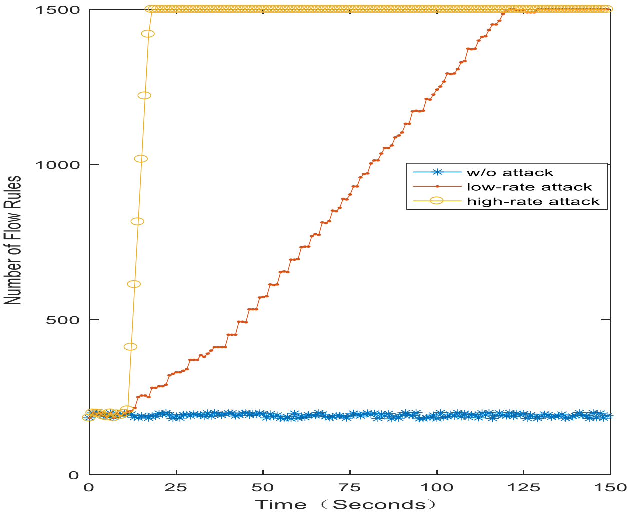


FIGURE 7. Number of flow rules in different network states.

As shown in Fig.7, in the normal network state, only the background traffic is running in the experimental network, and the number of flow rules is always maintained at about 180, which is relatively stable. In the network with high-rate DDoS attacks, the number of flow rules increased sharply in the tenth second when the attack was launched. Within less than ten seconds, the number of flow rules reached the upper limit of the switch. In the network with low-rate DDoS attacks, the attack also started from the 10th second, the flow table rules began to rise gradually, and it took more than 100 seconds for the number of flow rules to reach the upper limit of the switch. Once the number of flow tables reaches the upper limit, the flow rules of legitimate users cannot be installed in time. As shown in Fig.8, the number of FLOW_MOD messages is relatively small under normal network, sometimes FLOW_MOD messages are generated and sometimes they are not, which is an irregular trend. The trend of the number of FLOW_MOD messages in the network with low-rate DDoS attack is very similar to that in the normal network, which further proves the stealth of low-rate DDoS attack against the SDN data layer. In the network with high-rate DDoS attacks, we found that the number of FLOW_MOD messages increased sharply due to the attack, and the number of FLOW_MOD messages was always high. In our experiment, the number of FLOW_MOD messages was as high as 158 messages per second, until the final flow table was full. Since the packet rate of the low-rate DDoS attack is close to the packet rate of the normal data flow, and

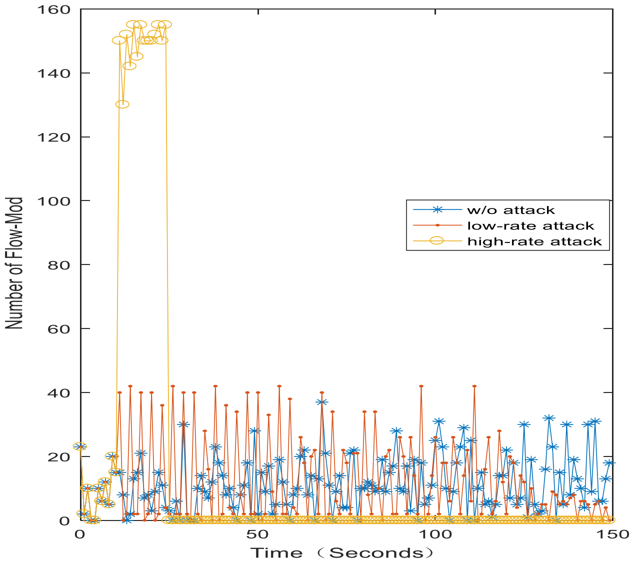


FIGURE 8. Number of FLOW-MOD messages in different network states.

the change caused by the flow table is small at the same time, the attack flow is concealed in the normal flow, making the detection more difficult. Therefore, we propose a DoS attack detection method based on FM algorithm.

2) MACHINE LEARNING PERFORMANCE INDICATOR

In order to verify the classification effect of FM machine learning, this experiment uses AUC (Area under Curve, AUC) as the main detection performance indicator while recall rate, precision and accuracy are the auxiliary indicators. In the field of statistics and machine learning, two aspects for the two-category problem are considered as follows.

i. The samples are divided into two categories, one is positive, which refers to abnormal traffic, and the other is negative, which refers to normal traffic.

ii. *TP*, *TN*, *FP*, *FN* correspond to the number of true positive classes, the number of true negative classes, the number of false positive classes, and the number of false negative classes.

According to *TP*, *TN*, *FP* and *FN*, the following indicators can be calculated.

a) Recall Rate:

$$recall = \frac{TP}{TP + FN} \tag{11}$$

It represents the proportion of samples that correctly predict the amount of attack traffic as a percentage of the total attack traffic sample.

b) Precision Rate:

$$precision = \frac{TP}{TP + FP} \tag{12}$$

It indicates that the number of samples correctly predicted as attack traffic accounts for the proportion of all predicted traffic samples.

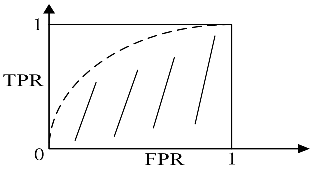


FIGURE 9. ROC schematic diagram.

c) Accuracy Rate:

$$accuracy = \frac{TN + TP}{TN + TP + FP + FN} \tag{13}$$

It represents the proportion of correctly classified samples to the total number of samples.

d) Receiver Operating Characteristic (ROC): The horizontal coordinate of the curve is error acceptance rate.

$$FPR = \frac{FP}{TN + FP} \tag{14}$$

Curve ordinate is recall rate.

$$TPR = \frac{TP}{TP + FN} \tag{15}$$

A schematic diagram of the ROC curve is shown in Fig.9. It is a series of binary methods, such as threshold division, with FPR as the abscissa and TPR ordinate.

e) Area Under Curve (AUC): The value of AUC is the shaded area of the ROC curve in Fig.8. It can be used to judge whether the model is good or bad, and any point on the curve can be used as the predictor threshold of the classifier.

The specific calculation formula for AUC is Equation 15.

$$AUC = \frac{\sum_{in\ s_i \in\ postiveclass} Rank_{in\ s_i} - \frac{M(M+1)}{2}}{M \times N} \tag{16}$$

Where $Rank_{ins_i}$ represents the sequence number of the i^{th} traffic sample (probability scores are sorted from small to large, ranked at the rank position); M and N represent the number of positive and negative samples, respectively; $\sum_{ins_i \in\ postiveclass}$ indicates that only positive sample numbers are summed.

3) FEATURES ANALYSIS

In the experiment, we set the window function size as 20s, collected and calculated duration time, packets number, RDMB, RDPI of each flow rule as shown in the next series of figures.

The duration time of each flow rule in the flow table is recorded by window function, and the mean and variance of the survival time of normal flow rules and attack flow rules are calculated. The results showed that within the window function of 20s, the average duration time of normal flow rule is about 10s, which verified the above literatures. However, the attack flow rule is kept in 20s, indicating that the attack flow rules always existed. It can be seen from the variance of survival time that the survival time of normal traffic is random, so the variance is large. However, as low-rate DDoS attack gradually fills the flow table, the flow rules of normal

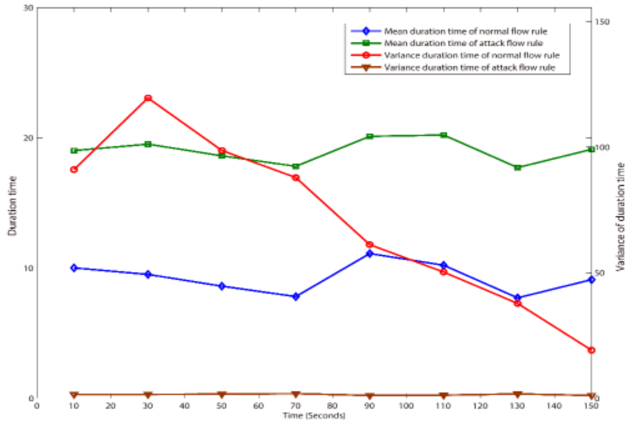


FIGURE 10. Variation of survival time eigenvalue.

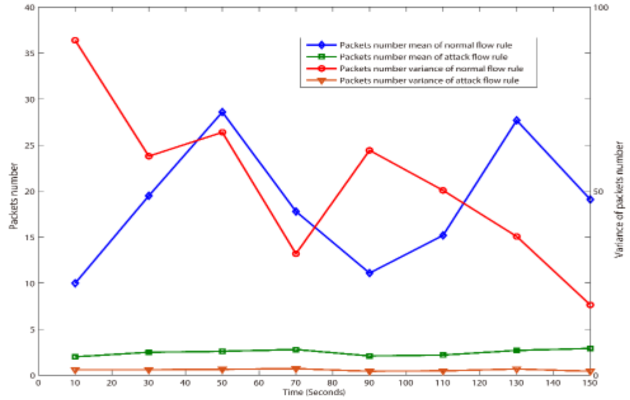


FIGURE 11. Variation of packets number eigenvalue.

traffic gradually decrease, and the variance also decreases. Because low-rate DDoS attack traffic is not random, all variances are small. As can be seen from Fig.10, the survival time of flow rules can well distinguish attack traffic from normal traffic.

It can be seen from Fig.11 that the number of matching packets of the normal flow rule is generally random, and the mean value is between 10 and 30 and the variance is large. As the flow table is exhausted by the low rate DDoS attack, the number of normal flow rule is decreasing and the variance is also decreasing. The number of packets matched by the attack flow rule in the window function is kept at about 2, and the variance is very low, which verifies the periodicity of the low-rate DDoS attacks.

From Fig.12, we can see that the mean and variance of the RDMB of the normal flow rule are kept at a very high value. This is due to the randomness of the normal flow, and the decrease of the variance is related to the gradual filling of the flow table. The RDMB mean and variance of the attack flow rule are kept at a very low value because the packet size of each flow under the low rate DDoS attack is small and constant. From the experiment we can see that RDMB can distinguish between normal flow rules and attack flow rules.

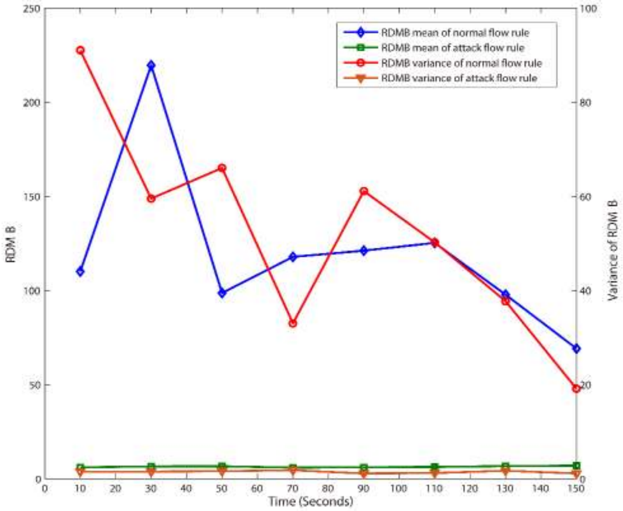


FIGURE 12. Variation of RDMB eigenvalue.

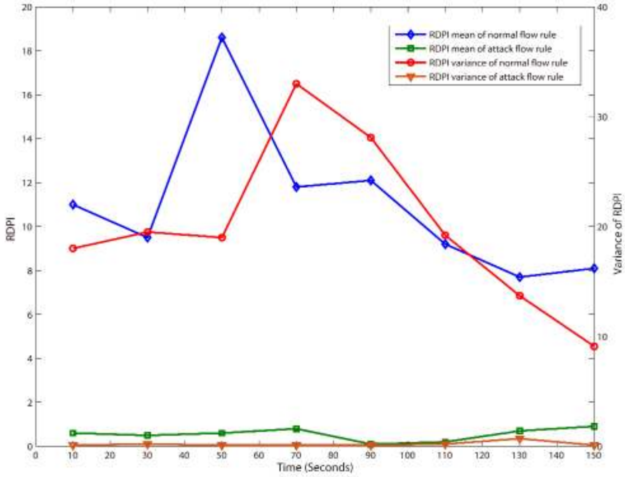


FIGURE 13. Variation of RDPI eigenvalue.

From Fig.13, we can see that the average RDPI of the normal flow rule is between 8 and 19, and the variance is between 6 and 16. The decrease of the RDPI variance is caused by the low rate DDoS attacks gradually filling the flow table, causing the number of normal flow rules become fewer. Due to the periodicity of low-rate DDoS attacks, the RDPI mean and variance of the attack flow rules are kept at a very low value. From the experiment, we can see that RDPI can distinguish between normal flow rules and attack flow rules.

4) PERFORMANCE ANALYSIS OF LOW-RATE DDoS ATTACK DETECTION METHOD BASED ON FM

Since FM machine learning algorithm requires a large number of training samples, we choose to collect a total of 140,000 sets of feature data and use the attack port to determine the data. Among them, 100,000 sets of features were selected as training sets, and 40,000 sets of features were selected as test sets.

TABLE 1. Detection performance under different parameters.

Learning rate γ hidden vector dimension k	recall	precision	accuracy	training time
0.1, 20	0.833	0.855	0.878	0.555s
0.1, 30	0.870	0.908	0.905	0.758s
0.2, 20	0.860	0.882	0.906	0.656s
0.2, 30	0.894	0.927	0.937	0.863s

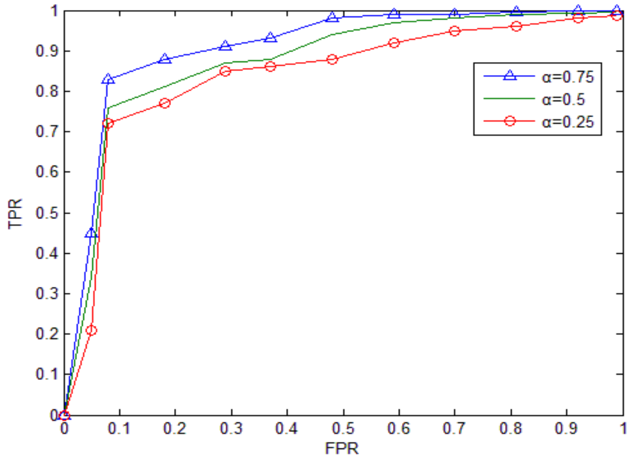


FIGURE 14. ROC curves at different attack rates.

In FM machine learning algorithm, the learning rate and the hidden vector dimension k are the parameters that have the greatest impact on their performance. Therefore, we selected four different combinations of parameters according to the literature [31] for the experiment, the average attack rate was 2.2 Mbps, and the performance comparison as shown in TABLE 1 was obtained.

From the experimental results in the table, we can see that when we set the learning rate to 0.2 and the hidden vector dimension k to 30, the detection effect of the FM-based DoS attack detection method is the best, so we will learn the rate and hide. The vector dimension k is set to 0.2 and 30, respectively. Therefore, the FM-based DoS attack detection method is designed. Below we will analyze the impact of low rate DDoS attack rate on detection methods.

We set the attack rate of the low rate DDoS to a different value for the test table. The average attack rate of low rate DDoS satisfies equation (17).

$$v_L = v_B * (1 + \alpha) \quad 0 < \alpha < 1 \tag{17}$$

where v_B is the average rate of background traffic, and α is set to 0.25, 0.5, and 0.75 respectively. The lower the α value, the closer the average attack rate of the low-rate DDoS is to the normal traffic. The attack traffic at this time is most similar to the normal traffic. The ROC de-curve of the detection method at different attack rates is shown in Fig.14.

TABLE 2. Comparison of performance at different attack rates.

α	recall	precision	accuracy	AUC
0.25	0.912	0.924	0.925	0.886
0.5	0.933	0.935	0.926	0.918
0.75	0.946	0.950	0.958	0.938

TABLE 3. Performance comparison of different benchmark dataset.

dataset	recall	precision	accuracy	AUC
NSL-KDD	0.932	0.928	0.925	0.925
DARPA98	0.952	0.949	0.932	0.916
CAIDA	0.946	0.950	0.958	0.938

The overall performance of the detection method under different attack rates is shown in TABLE 2.

It can be analyzed from TABLE 2 that the attack rate is close to the average rate of background traffic, and the detection performance is worse. Because the attack traffic is similar to the background traffic, the attack traffic is difficult to distinguish from the background traffic, so the detection performance is degraded.

In order to verify the good performance of FM algorithm, the performance of FM algorithm is also compared with benchmark data sets such as NSL-KDD and DARPA98, the experimental results are shown in TABLE 3.

It can be seen from the TABLE 3 that no matter what kind of data set, FM algorithm has better performance. This is enough to prove that FM algorithm has a better detection effect under different benchmark data sets.

5) COMPARATIVE ANALYSIS OF DIFFERENT MACHINE LEARNING ALGORITHMS

In order to verify the advantages of the detection algorithm in this paper, we also implemented two other machine learning algorithms in the experimental environment, and fixed the average attack rate to 2.2 Mbps ($\alpha = 0.75$). Their performance comparison is shown in TABLE 4.

Through the comparison experiments of different machine learning algorithms, we can see that FM algorithm is superior to CNN and Random Forest in terms of various detection indicators. This is because FM algorithm introduces the feature combination mechanism, and the feature combination detection method can establish the correlation between each feature sample, so that the feature samples actually used to update the parameters are more abundant. This is enough to prove that the DDoS attack detection method based on FM algorithm is better than other machine learning algorithms in detection performance.

6) PERFORMANCE COMPARISON OF DIFFERENT DETECTION METHOD

In order to verify the advantages of the detection method in this paper, under the same experimental environment

TABLE 4. Performance comparison of different machine learning algorithms.

Machine learning algorithm	recall	precision	accuracy	AUC
CNN	0.898	0.900	0.909	0.906
Random Forest	0.867	0.889	0.903	0.891
FM	0.946	0.950	0.958	0.938

TABLE 5. The potency comparison of different detection methods

Detection method	recall	precision	accuracy	AUC
Joint features	0.886	0.720	0.801	0.883
SDCC	0.870	0.775	0.825	0.836
This paper	0.946	0.950	0.958	0.938

condition (the average attack rate is fixed at 2.2 Mbps, i.e. $\alpha = 0.75$), we compare and analyze the detection method in this paper with the detection method based the joint features [17] and SDCC [9]. The evaluation criteria are designed and calculated according to the comparison model, mainly include the recall rate, precision rate and accuracy rate. The comparison results are shown in TABLE 5.

As can be seen from TABLE 5, the recall of the detection method based the joint features [17] reaches 88.6%, the accuracy reaches 80.1%, and the AUC reaches 88.3%. The recall of SDCC [9] reaches 87.0% and the accuracy reaches 82.5%. The recall of DDoS attack detection method based on FM algorithm proposed in this section is 94.6%, the accuracy is 95.8%, so the detection effects are obviously improved, and all are better than the former two methods. This is because the available bandwidth percentage among the joint features proposed in [17] may have certain impacts on the detection effects. When the attack is in the initial stage, the network bandwidth occupancy rate will gradually increase. In this case, the discrimination between the bandwidth occupancy rate under attacked and the normal bandwidth occupancy rate is not obvious enough. There are some deficiencies in the feature extraction method of the attack traffic. However, this paper carries out modeling detection aims at the behavior and communication characteristics of a single attack flow, and the established model is less affected by the overall traffic characteristics, so it can effectively detect low-rate DDoS attacks. Compared with the detection method aims at the overall attack traffic, the detection effects are significantly improved.

In paper [9], as the weight of the attribute value in the SDCC algorithm needs to be set manually according to the actual situation, this may lead to certain errors in the detection results. In order to improve the data processing rate, the SDCC scheme will extract different proportions of data for analysis in different detection stages. When there is a sudden change in normal traffic, only part of the data is extracted for analysis may lead to some misjudgments. However, the detection range of this paper will cover all traffic, and it will analyze and detect the behavior and communication characteristics of each flow to ensure the accuracy of detection. Meanwhile, by collecting statistical information of the flows through the controller API, which is closely combined with the SDN framework, to ensure the higher efficiency of the entire detection method.

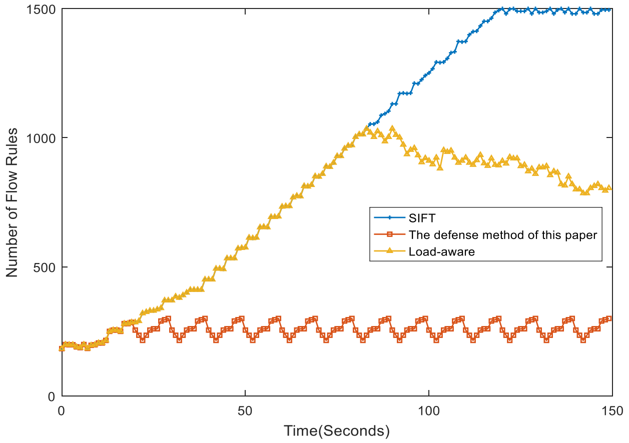


FIGURE 15. Changes in the number of flow rules in different defense methods.

7) DEFENSE METHOD BASED ON DYNAMIC DELETION OF FLOW RULE

Run the experimental network and inject the background traffic. In the 10th second, the low-rate DDoS attack is initiated from h1 to h9, and the detection and defense functions are enabled on the controller. The time interval t of dynamically deleting the flow rule is set to 10 seconds. In order to verify the advantages of the DDoS attack defense method based on dynamic deletion of flow rules, a comparative experiment was conducted between the method proposed in this paper and the method proposed in literature [11] and literature [29] during the simulation process.

From Fig.15, we can see that in the first ten seconds without attack, the number of flow rules remains within a stable range. When the attack is launched, the number of flow rules starts to rise in a gradient. When the DDoS attack defense method based on the dynamic deletion of flow rule is adopted, the flow rules are deleted every ten seconds. This method makes the number of flow rules always around 350, which can effectively control the growth of flow rules in the switch, so that the low-rate DDoS attack cannot achieve the purpose of occupying the flow table capacity. However, the SIFT-based defense method in literature [29] starts the defense strategy only when the flow table is saturated. The execution process of this strategy is that once the flow table is saturated, the controller will randomly delete flow rules, and the average number of flow rules deleted each time

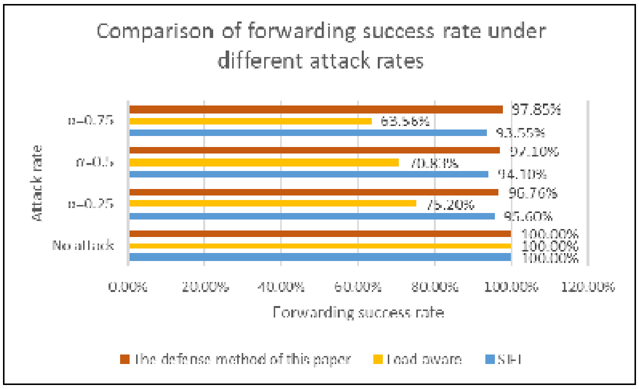


FIGURE 16. Comparison of forwarding success rate under different attack rates.

is 10, otherwise no defense strategy is adopted. Due to the continuous attack, there is still not enough space left in the flow table when the defense strategy is adopted. Therefore, the SIFT-based defense strategy cannot fundamentally solve the problem of flow table saturation under low-rate DDoS attacks. Literature [11] proposed a load-aware method that dynamically adjusts the timeout of flow rules by monitoring the state of the flow table. When the number of flow rules exceeds 60% of the flow table capacity, the controller will dynamically adjust the timeout every 10 seconds, which results in the premature disappearance of some flow rules and reduces the number of flow rules in the switch. However, the continuation of the attack will lead to an increase in the number of flow rules. Therefore, the number of flow rules in the flow table generally shows a dynamic equilibrium state. Although this method prevents the flow table from being saturated to a certain extent, the method still has certain defects.

In order to further prove the advantages of the defense method of this paper, we also compare the forwarding success rate of normal data packets (refers to the ratio of the normal packets that are forwarded successfully to the total number of normal data packets after taking defensive measures) under the three defense methods.

As can be seen from Fig.16, the forwarding success rate of normal data packets is higher after adopting the defensive method in this paper. On the one hand, the deletion of flow rules is based on accurate detection, which can greatly reduce the probability of accidental deletion of legitimate flow rules. On the other hand, due to sufficient remaining space, more normal clients can be forwarded, so the forwarding success rate of normal packets is higher. After adopting the SIFT-based defense method, the controller will randomly delete the flow rules. On the one hand, it is easy to cause the accidental deletion of legal flow rules. On the other hand, the insufficient space in the flow table will lead to a large number of normal clients not getting the response, thus reducing the success rate of forwarding normal packets. However, the load-aware method controls the growth of the number of flow rules by dynamically adjusting the timeout. As the timeout decreases, more legitimate flow rules will be deleted prematurely, resulting in a lower forwarding success rate of normal packets.

In addition, after adopting the SIFT-based defense method, as the attack rate increases the forwarding success rate of normal packets decreases. This is because the higher the attack rate, the more likely the number of flow rules will reach the upper limit of the flow table capacity, resulting in more legal flow rules getting no response. After adopting the load-aware approach, as the attack rate increases, the forwarding success rate of normal packets also decreases. This is because the higher the attack rate, the faster the timeout decreases, which will cause more deletion of legal flow rules. However, after adopting the defense method in this paper, as the attack rate increases, the success rate of normal data packets increases. This is because the higher the attack rate, the better the detection effect, so the flow rules can be deleted more accurately, which improves the forwarding success rate of normal packets.

V. CONCLUSION

In recent years, as a new type of network architecture, software-defined networks have attracted great interest from researchers. They are gradually being widely applied to various fields of the network. However, low-rate DDoS attacks against the data layer have not yet become research hotspots, and related research results are also less. This paper first studies how to launch such attacks and verifies the effectiveness of such attacks. Then, by extracting the four features related to the flow rules, the feature data set for detecting such attacks is established, and the FM-based detection method for low-rate DDoS attack in SDN is proposed. By comparing the detection performance under different machine learning algorithms, and comparing with the existing low-rate attack detection methods, it is verified that the DDoS attack detection method based on FM algorithm has higher recall rate, precision rate and AUC value. Good detection performance provides a reliable condition for defending against such attacks. Finally, based on the detection, this paper proposes a defense method based on dynamic deletion of flow rules, which effectively prevents the flow table from being saturated, and improves the forwarding success rate of normal data packets, thus ensuring the normal operation of SDN.

In the future work, we plan to use more evaluation criteria to compare the proposed scheme with more deep learning methods. In addition, in order to further prove the effectiveness of the proposed scheme, we plan to deploy the proposed scheme to a real network environment.

REFERENCES

[1] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[2] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 602–622, 1st Quart., 2016.

[3] S. Yu, Y. Tian, S. Guo, and D. O. Wu, "Can we beat DDoS attacks in clouds?" *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 9, pp. 2245–2254, Sep. 2014.

[4] F. Hu, Q. Hao, and K. Bao, "A survey on software-defined network and openflow: From concept to implementation," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 2181–2206, 4th Quart., 2014.

[5] S. Scott-Hayward, S. Natarajan, and S. Sezer, "A survey of security in software defined networks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 1, pp. 623–654, 1st Quart., 2016.

[6] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: Scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*, 2013, pp. 413–424.

[7] M. Dhawan, R. Poddar, and K. Mahajan, "SPHINX: Detecting security attacks in software-defined networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, Feb. 2015, pp. 8–11.

[8] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Feb. 2015, pp. 77–81.

[9] H. He, Y. Hu, and L. H. Zheng, "Efficient DDoS attack detection and prevention scheme based on SDN in cloud environment," *J. Commun.*, vol. 39, no. 4, pp. 139–151, 2018.

[10] M. Liu, *Research on DDoS Attack and Defense System and Key Technologies in Cloud Environment*. Nanjing, China: Nanjing University, 2016.

[11] H. Zhu, H. Fan, X. Luo, and Y. Jin, "Intelligent timeout master: Dynamic timeout for SDN-based data centers," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 734–737.

[12] Y. Wang, T. Hu, G. Tang, J. Xie, and J. Lu, "SGS: Safe-guard scheme for protecting control plane against DDoS attacks in software-defined networking," *IEEE Access*, vol. 7, pp. 34699–34710, 2019.

[13] K. Kalkan, G. Gur, and F. Alagoz, "SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment," in *Proc. IEEE Symp. Comput. Commun.(ISCC)*, Jul. 2017, pp. 669–675.

[14] K. Kalkan, L. Altay, G. Gur, and F. Alagoz, "JESS: Joint entropy-based DDoS defense scheme in SDN," *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2358–2372, Oct. 2018.

[15] C. Xu, H. Lin, Y. Wu, X. Guo, and W. Lin, "An SDNFV-based DDoS defense technology for smart cities," *IEEE Access*, vol. 7, pp. 137856–137874, 2019.

[16] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proc. IEEE Local Comput. Netw. Conf.*, Oct. 2010, pp. 408–415.

[17] Z. J. Wu, J. A. Zhang, and M. Yue, "Approach of detecting low-rate DoS attack based on combined features," *J. Commun.*, vol. 38, no. 5, pp. 19–30, 2017.

[18] C. Bu, X. Wang, M. Huang, and K. Li, "SDNFV-based dynamic network function deployment: Model and mechanism," *IEEE Commun. Lett.*, vol. 22, no. 1, pp. 93–96, Jan. 2018.

[19] L. Barki, A. Shidling, N. Meti, D. G. Narayan, and M. M. Mulla, "Detection of distributed denial of service attacks in software defined networks," in *Proc. Int. Conf. Adv. Comput., Commun. Informat. (ICACCI)*, Sep. 2016, pp. 2576–2581.

[20] N. Meti, D. G. Narayan, and V. P. Baligar, "Detection of distributed denial of service attacks using machine learning algorithms in software defined networks," in *Proc. Int. Conf. Adv. Computing, Commun. Informat. (ICACCI)*, Sep. 2017, pp. 1366–1371.

[21] S. Garg, K. Kaur, N. Kumar, and J. J. P. C. Rodrigues, "Hybrid deep-learning-based anomaly detection scheme for suspicious flow detection in SDN: A social multimedia perspective," *IEEE Trans. Multimedia*, vol. 21, no. 3, pp. 566–578, Mar. 2019.

[22] S. Haider, A. Akhunzada, G. Ahmed, and M. Raza, "Deep learning based ensemble convolutional neural network solution for distributed denial of service detection in SDNs," in *Proc. UK/China Emerg. Technol. (UCET)*, Aug. 2019, pp. 1–4.

[23] J. Wu, M. Dong, K. Ota, J. Li, and Z. Guan, "Big data analysis-based secure cluster management for optimized control plane in software-defined networks," *IEEE Trans. Netw. Service Manag.*, vol. 15, no. 1, pp. 27–38, Mar. 2018.

[24] Y. G. Dantas, V. Nigam, and I. E. Fonseca, "A selective defense for application layer DDoS attacks," in *Proc. IEEE Joint Intell. Secur. Informat. Conf.*, Sep. 2014, pp. 75–82.

[25] E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello, "Mobile executions of slow DoS attacks," *Log. J. IGPL*, vol. 24, no. 1, pp. 54–67, 2016.

[26] E. Cambiaso, G. Papaleo, G. Chiola, and M. Aiello, "Designing and modeling the slow next DoS attack," *Int. Joint Conf.*, vol. 369, pp. 249–259, Jun. 2015.

[27] K. Kannan and S. Banerjee, "Compact TCAM: Flow entry compaction in TCAM for power aware SDN," in *Proc. ICDCN*, vol. 7730, 2013, pp. 439–444.

[28] R. Kandoi and M. Antikainen, "Denial-of-service attacks in openflow SDN networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 1322–1326.

[29] T. A. Pascoal, Y. G. Dantas, and I. E. Fonseca, "Slow TCAM exhaustion DDoS attack," in *Proc. 32nd IFIP Int. Conf.*, 2017, pp. 17–31.

[30] S. Kandula, S. Sengupta, and A. Greenberg, "The nature of data center traffic: Measurements & analysis," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, Nov. 2009, pp. 202–208.

[31] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2011, pp. 995–1000.

[32] *Mininet*. Accessed: Jun. 2018. [Online]. Available: http://www.mininet.org/

[33] (2016). *Ryu*. [Online]. Available: https://osrg.github.io/ryu/

[34] A. Botta, A. Dainotti, and A. Pescapé, "A tool for the generation of realistic network workload for emerging networking scenarios," *Comput. Netw.*, vol. 56, no. 15, pp. 3531–3547, Oct. 2012.

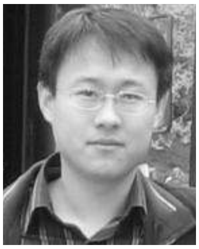
[35] CAIDA Datasets. (2007). *DDoS Attack*. Accessed: 2018. [Online]. Available: https://data.caida.org/datasets/security/ddos-20070804

[36] *Scapy*. Accessed: May 2017. [Online]. Available: http://www.secdev.org/projects/scapy

WU ZHIJUN received the B.S. and M.S. degrees in information processing from Xidian University, China, and the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, China. He was a Professor with the Department of Communication Engineering, Civil Aviation University of China. His research interests include denial-of-service attacks, and security in big data and cloud computing.

XU QING received the B.S. degree in electronic information engineering from Chuzhou University, China. He is currently pursuing the master's degree in network security with the Civil Aviation University of China. His main research interest includes denial-of-service attacks and security in SDN.

WANG JINGJIE received the B.S. degree in electronic information engineering and the M.S. degree in information security from Civil Aviation University, China. His research interest includes denial-of-service attacks.



YUE MENG received the B.S. degree in electronic information engineering from the Hebei University of Science and Technology, China, and the M.S. degree in information security from the Civil Aviation University of China, China. He was a Lecturer with the Department of Communication Engineering, Civil Aviation University of China. His research interest includes denial-of-service attacks.

LIU LIANG received the B.S. degree in electronic information engineering from the Harbin Institute of Technology, China, and the M.S. degree in information security from the Civil Aviation University of China, China. His research interest includes denial-of-service attacks.