IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 23, 2020, accepted May 30, 2020, date of publication June 3, 2020, date of current version June 16, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2999668

Detecting DoS Attacks Based on Multi-Features in SDN

MENG YUE, HUAIYUAN WANG, LIANG LIU, AND ZHIJUN WU
School of Electronics, Information and Automation, Civil Aviation University of China, Tianjin 300300, China

Corresponding author: Meng Yue (myue_23@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61601467 and Grant U193310081, in part by the Fundamental Research Funds for the Central Universities of CAUC under Grant 3122018C003, and in part by the Scientific Research Project of Tianjin Municipal Education Commission under Grant 2019KJ117.

ABSTRACT Denial of Service (DoS) attack is a serious threat to Software Defined Network (SDN). Although many research efforts have been devoted to identify new features for DoS attack detection, the existing approaches are not able to detect various types of DoS attacks. In SDN, DoS attacks against data plane are mainly organized in two ways: 1) DoS attack with multiple flow entries (M-DoS) to exhaust the Ternary Content-Addressable Memory (TCAM) resource of the switch. 2) DoS attack with a single well-designed entry (S-DoS) to overwhelm the target link and further impact the controller. To detect these two attacks, we propose a new approach by extracting six features of flow table, and using the back propagation (BP) neural network to construct the classifier. Test results of test-bed experiments indicate that the accurate detection probability of proposed approach is 98.9%, which can effectively distinguish M-DoS flows and S-DoS flows without being affected by Flash crowd scene.

INDEX TERMS SDN security, DoS, feature detection, flow table, flash crowd.

I. INTRODUCTION

Software Defined Network (SDN), as a new type of network management architecture, provides network with flexible control, simple network architecture, and great programmability by decoupling the control plane and the data plane of the traditional network. The control plane enables upper-level managers to implement the required functions by simply deploying the network. In terms of development, SDN provides developers with rich programming interfaces, enabling them to change the network deployment according to actual needs. The logically centralized controller not only provides powerful technical support for complex network services, but also can obtain network status information from a global perspective, which is convenient for monitoring the network in real-time. Moreover, the separation of the control plane and the data plane simplifies the process of packet forwarding, reduces the load on the switch, and makes network configuration more convenient. SDN has been widely used in network virtualization, wireless LANs, cloud computing and other fields due to its advantages [1]–[4]. However, as SDN is still in the development stage, many technical details are not mature enough, it can easily become a key target of network attacks. SDN has serious security vulnerabilities, thus faces great security threats [5]–[7].

Denial of Service (DoS) attacks is one of the major threats to SDN. Previous researches have noted that there exist many types of DoS attacks in SDN [8]–[10]. The victim may be control plane, data plane or application [9], [11]. DoS attacks against data plane, control plane or SDN application commonly have different principles and features, corresponding to specialized detection methods. Therefore, DoS attacks should be studied according to different planes, and existing studies are commonly conducted in this way. Currently, DoS attacks on data plane have attracted wide attention, as the vulnerabilities of this plane are exposed. These attacks can be classified into M-DoS (e. g. the flow table-oriented attack, Table Overflow [17]) and S-DoS (e. g. the bottleneck link oriented attack, Crosspath [23]). Although many efforts have been devoted to these two attacks, we found two shortcomings in existing studies: 1) high false positive rate in Flash crowd scenario; 2) no approach could detect both M-DoS and S-DoS. Especially, the lack of effective detection of S-DoS. These two shortcomings motivated us to work on the data plane. M-DoS and S-DoS attacks against data plane are shown in Figure.1.

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

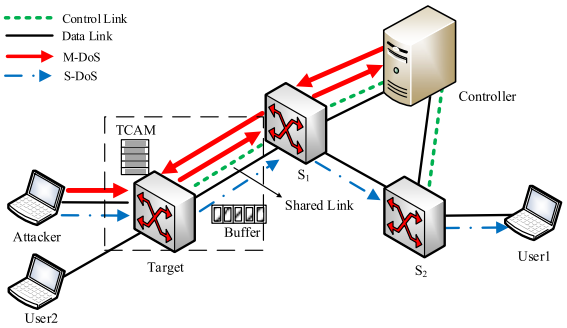


FIGURE 1. The attack diagram.

1) The attacker exhausts the Ternary Content-Addressable Memory (TCAM) of the target switch by triggering multiple new flow entries (M-DoS) [12], [13]. OpenFlow protocol is commonly used to define the communication regulations between the controller and the switch [14]. According to OpenFlow, when the first packet of a new flow (a flow is defined as a set of data packets forwarded by the same flow entry in the flow table) arrives at the switch, a *packet_in* message will be sent to the controller. The controller then responses a *packet_out* message to the switch to install a new rule in its flow table. Because of the growing demands for a fast and efficient data plane, flow tables are usually implemented using TCAM. TCAMs are expensive and have high power consumption. Therefore, SDN switches have a limited TCAM space that can store only a limited number of flow rules [15], [16]. In this case, M-DoS attacks can be launched by forcing a target switch to install a great number of fake flow rules [17], [18].

2) The attacker launches a single well-designed entry to congest the shared link (S-DoS) [20], [21]. The recently proposed CrossPath attack can be regarded as a typical S-DoS attack. The shared links between paths of control traffic and data traffic in SDN is a common practice in SDN with in-band control [22], which can greatly reduce the cost of building a dedicated control network and simplify network maintenance, especially for large networks. In this scenario, the attacker can send well-designed data traffic to the shared link to interfere the control traffic instead of directly overwhelming the controller. Once the shared link buffer is congested, real-time control messages delivered between the controller and the switch are significantly delayed or dropped. Furthermore, in order to ensure that attack packets can quickly occupy the shared link, the attacker generally does not inject new flow entries, but uses attack flows that match the same flow rule, thereby avoiding the delay that results from the installation of new flow rules [23], [24]. S-DoS attacks are concealed and can evade existing methods for detecting M-DoS attacks.

So far, many studies mainly focus on M-DoS attacks detection in SDN [25]–[27]. The detection of S-DoS attacks has not yet attracted much attentions. In addition, most of the existing M-DoS detection approaches fail to detect S-DoS attacks. Also, they are generally difficult to accurately distinguish between M-DoS attacks and Flash crowd, since M-DoS attacks and normal Flash crowd have similar behaviors. In order to overcome the aforementioned shortcomings, we propose a novel attack detection approach based on multi-features, and test its performance. The primary contributions of this paper can be summarized as follows:

a) We extract six new features of flow table for DoS attack detection, which can accurately reflect the behaviors of M-DoS attack, S-DoS attack and Flash crowd. The combined usage of these features can improve the resolution.

b) We design a classifier based on back propagation (BP) neural network to distinguish between M-DoS attacks, S-DoS attacks and Flash crowd. Importantly, our approach outperforms other detection approaches in terms of higher detection rate and lower false positive rate.

The remaining of this paper is organized as follows. Section 2 introduces related works. Section 3 describes the proposed detection approach based on multi-features. Section 4 presents results from test-bed experiments to evaluate the detection performance of the proposed approach. Section 5 concludes this paper and discusses future works.

II. RELATED WORKS

SDN has advantages in traffic control and network management, but it is vulnerable to DoS attacks. How to defeat such attacks in SDN has attracted wide attention. There are generally two detection methods for DoS attack in SDN, the detection method based on threshold and the detection method based on features. The detection method based on threshold generally monitors network behaviors in real time. Once a certain indicator exceeds the pre-set threshold, the attack is considered to occur. Dhawan *et al.* [25] proposed a DoS detection approach by monitoring the installation speed of flow rules. If the speed exceeds a certain threshold, the network may be attacked and then trigger defense mechanisms. Chou *et al.* [26] analyze the correlation between links in the network, and determine whether there is an attack in the network by measuring the time difference between the round-trip times of each Link Layer Discovery Protocol (LLDP) frame. Liu *et al.* [27] uses the generalized entropy method to detect the traffic on the switch, find the abnormal switch based on whether the generalized entropy value exceeds the threshold, and then use the neural network for further analysis. However, such detection usually depends on only a few indicators, so it is easy to mistake normal random bursts in practical networks as attacks. Moreover, the outcomes of these approaches require very high accuracy of threshold. The threshold will be varied correspondingly by the change of the network scenario, otherwise the accurate detection probability will be affected seriously. The essence of feature-based detection is to establish a classifier to classify the normal flow and the attack flow. Generally, statistical analysis, neural network, support vector machines and other methods are used to process attack features and further build detection model. For DoS attacks, Kirutika *et al.* [28] proposes a

TABLE 1. The summary of detection method.

Detection method	Advantages	Disadvantages
Threshold-based methods[25]-[27]	• Simple implementation • Real-time detection	• Single threshold results in high false positive • Difficult to determine the threshold • Impacted by network parameters
Feature-based methods [28]-[30]	• High detection rate and low false positive rate • Robust in different network scenarios	• Need fine-grained feature extraction • Most of features need to pre-process

monitoring system based on the behavior of the controller, calculates the probability of being attacked by monitoring the behavior of the controller, and uses the random forest algorithm to improve the accuracy of the intrusion detection system. Wang and Chen [29] proposed a SDN Safe-Guard (SGuard) architecture to identify attacks from normal traffic. Such classification module is composed of data collector, feature extractor and Self Organizing Maps (SOM) classifier. They evaluate the performance of SGuard in Mininet, which shows SGuard was lightweight and efficient. Prasath and Perumal [30] proposes a detection method based on Bayesian algorithm which is used to train historical network attack data. The method can predict the attacking host and prevent it from continuing to connect. Experimental results show that the prediction using Bayesian network has a high accuracy. The summary of the two types of detection methods is shown in Table 1.

Compared to the threshold-based detection, the feature-based detection commonly has better performance in detection rate and robust. Inspired by the feature-based detection and the global view and centralized control of the controller, we proposed our detection method based on Multi-features in SDN. We deploy the detection method on the controller, obtain the switch flow table information through the controller. Then extract the feature from the flow table, which as the inputs of the BP neural network. Finally, the network traffic is classified into the M-DoS traffic, the S-DoS traffic and the normal traffic according to the outputs of the BP neural network.

III. DoS DETECTION BASED ON FLOW TABLE FEATURES

The feature-based detection requires extensive collection of anomalous features within the network. This is very easy under SDN because the controller is the information center of the network. The flow table of SDN switches stores a large amount of flow information, from which some useful features can be extracted for DoS detection. Moreover, the performance of the feature-based detection mainly depends on the number of features and the representativeness of features: 1) Too few features can easily lead to inaccurate detection while too many features will increase overhead. 2) The representativeness of features means that the extracted features are significantly different from normal flow features. The flow table in the switch records the information of each entry that flows through the switch. It can globally reflect the changes of flow entries with time, and its versatility is better, which does not need to consider the specific application layer protocol. For the exhausted TCAM attack and congested bottleneck link attack we studied, the characteristics of the flow table are significant. Therefore, we detect DoS attacks based on the characteristics of the flow table. In this section, we extract six features from SDN flow tables, and then use a BP neural network classifier to design the detection model, which can accurately identify M-DoS flows, S-DoS flows and normal flows.

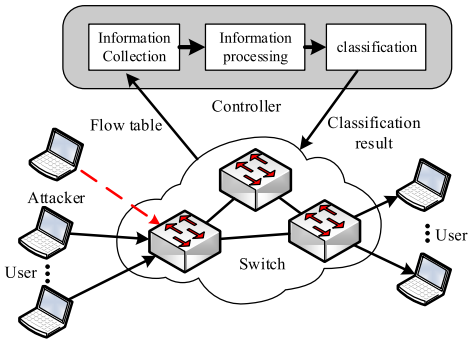


FIGURE 2. Detection architecture.

A. CLASSIFIER METHOD ARCHITECTURE

The programming features of SDN enable us to conveniently deploy the detection method on the controller. In terms of the architecture, our detection method is divided into three modules: information collection, information processing and classification, as shown in Figure. 2.

In Figure 2, the information collection module uses the OpenFlow API of the controller to request flow table information from the switch at a suitable cycle. The information collection module sends the collected flow table to the information processing module, where the features were extracted from flow table. Each entry of flow information in the flow table is processed into a six-tuple consisting of six features. The six-tuple is input to the BP classifier that has been trained in the classification module. The classification module determines whether the network is under attack based on the output of the classifier.

B. DEFINITION OF ATTACK FEATURES

Suitable features are the key to our detection method based on multi-features. Primarily, four new representative features are extracted which are sensitive to different DoS attacks. The former two present the property of M-DoS, while the latter two show the property of S-DoS. Then, we propose two more features to distinguish DoS attack and Flash crowd, reducing the rate of false alarm.

1) ESIPs (ENTROPY OF SOURCE IPs)

Previous work [28] provided a solution for detecting M-DoS attacks based on the entropy variation of the destination IP address. However, this solution is not applicable to detect the DoS targeting at much more destination IP addresses (e.g. Ip sweep attack). Because the destination IP address is more dispersed, the entropy value will be higher which results in a low detection probability. Here, we proposed the source-Ip-based entropy to detect M-DoS based on the fact that the attacker usually uses IP source spoofing to construct massive new packets and makes attacks hard to detect. We assume n represents the number of flow entries in a flow table, k represents the number of source IP addresses in a flow table, and x_i represents the number of flow entries corresponding to the i-th source IP. The Entropy of Source IPs (ESIPs) is defined as follows.

$$H(x) = -\sum_{i=0}^{k} p_{x_i} \lg p_{x_i} \tag{1}$$

where $p_{xi} = x_i/n$ represents the probability of flow entries corresponding to the i-th source IP. When the network suffers M-DoS attacks, the value of ESIPs will increase dramatically because the source IPs are more dispersed.

2) SFT (SIMILARITY OF FLOW TABLES)

The ESIPs only considers one element, the source IP. Here, we further consider the correlation of flow tables over time, which is a global consideration. In normal condition, the flow table has such a stable distribution of flow entries that flow table will not change a lot over time. However, the M-DoS attack will cause a lot of new flow entries stored in the flow table, thus the flow table will change over time. We let fe denote a flow entry including a five-tuple: protocol number, source/destination port and source/destination IP. We let fc composed of $\{fe_1, fe_2, \ldots, fe_n\}$ represent the collection of the total flow entries in a flow table, which is a set of fe. The SFT is defined as follows:

$$SFT = F_{SFT}(fc_i, fc_{i+1}) \tag{2}$$

where fc_i represents the flow entries set collected at the i-th moment, F_{SFT} is a function for calculating the similarity of fc_i and fc_{i+1}. The algorithm of function F_{SFT} is shown as follows:

According the algorithm of function F_{SFT}, the value of SFT will be lower with the occurrence of M-DoS attacks.

3) GRMMP (GROWTH RATE OF MAX MATCHED PACKETS)

For S-DoS, we define the GRMMP to reflect its feature. OpenFlow provides a counter *n_packet* to record the number of matched packets for each flow entry. When the attacker launches S-DoS attacks, an attack flow entry will appear. Moreover, the number of packets forwarded by this flow entry will increase rapidly because massive attack packets are matched to this flow entry. We define the GRMMP as follows:

$$GRMMP = \frac{|MP_{i+1} - MP_i|}{\Delta t} \tag{3}$$

where $MP_i = \max(n\_packet_1, n\_packet_2, \ldots n\_packet_n)$ denotes the max *n_packet* in current flow table that stored n flow entries. MP_{i+1} denotes the max *n_packet* of the flow table after a sampling interval Δt. When an attacker launches the S-DoS attack, the number of packets matched by the attack flow entry increases dramatically, causing the GRMMP keep a higher value.

Algorithm 1 Calculating Similarity of Flow Table

Function F_{SFT} (fc_i, fc_{i+1})
Input: flow collection fc_i and flow collection fc_{i+1}
Return: the similarity of fc_i and fc_{i+1}
  $l = length(fc_i) + length(fc_{i+1}), fc_s = shoter(fc_i, fc_{i+1}), fc_l = longer(fc_i, fc_{i+1}), similarity\_count = 0$
  For fe_i in fc_s and $i = 1$ to $length(fc_s)$ **do**
    For fe_j in fc_l and $j = 1$ to $length(fc_l)$ **do**
      If $fe_i = fe_j$ then
        $similarity\_count = similarity\_count + 1$
      End if
    End for
  End for
Output: $similariy = \frac{2^* similarity\_count}{l}$
End function

4) GRMMB (GROWTH RATE OF MAX MATCHED BYTES)

Similar to the feature of GRMMP, GRMMB is designed to detect S-DoS attack too. In the flow table we collected, *n_byte* record the number of bytes that each flow entry matches. When the attacker launches S-DoS attacks, a growing number of attack packets belonging to a specific flow entry flood into the victim host through the switch, causing the *n_byte* of that flow entry increased rapidly. We define the GRMMB as follows:

$$GRMMB = \frac{|MB_{i+1} - MB_i|}{\Delta t} \tag{4}$$

where $MB_i = \max(n\_byte_1, n\_byte_2, \ldots n\_byte_n)$ denotes the max *n_byte* in current flow table stored n flow entries. MB_{i+1} denotes the max *n_byte* in next flow table after a sampling interval Δt. S-DoS attack produces lots of packets, and the number of matching bytes for the flow entry increases significantly, resulting in a large increase in the GRMMB value. Some S-DoS attack packets have large number but small size, and the other S-DoS attack packets do not have too much number but each packet has a lot of bytes to occupy resources. The combination of GRMMP and GRMMB will have a good detection of these S-DoS attacks.

5) PFSP (PERCENTAGE OF FLOWS WITH SMALL NUMBER OF PACKETS)

Flash crowd is a special kind of scene, a lot of legitimate users suddenly access the server within a short time

period, installing a large number of flow rules. Flash crowd behaves similarly to M-DoS attacks and is difficult to detect. Thus, distinguish Flash crowd and M-DoS attack become the key problem of attack detection technology. To solve the problem, we analyze real network data from CAIDA Datasets [33], [35]. In our analysis, we found that although both Flash crowd and M-DoS install lots of flow rules, the number of matched packets matched by their flow entries are different: the number of packets under M-DoS attack usually stays between 1 and 2, is no more than 3, while the number of packets in Flash crowd scene usually stays between 5 and 20. This may because that M-DoS attacks are mostly spoofing IP and no reply data exists. Thus, we design PFSP to distinguish Flash crowd and M-DoS attacks, which is defined as follows:

$$PFSP = \frac{\sum_{i}^{FlowSum} Flow_i(n\_packet < Threshold)}{FlowSum} \quad (5)$$

where *n_packet* remarks the number of matched packets for each flow entry. Threshold is used to filter flows under attacks, we set *threshold* to 3 in our experiment. $Flow_i$ is the number of attack flows after traverse all flow entries. We let $Flow_i$ divide sum of flow entries to represent the percentage of flows with small number of packets in the flow table. The PFSP will increase rapidly under M-DoS attacks, while not change in the Flash crowd scene.

6) PFSD (PERCENTAGE OF FLOWS WITH SHORT TIME DURATION)

After the analysis of CAIDA Datasets, we found except the number of matched packets of flow entry, the duration of the flow entry can also distinguish the Flash crowd and M-DoS attacks. In the Flash crowd, although the access is short and sudden, but it is legitimate and the source IP is genuine. Thus, the access will be returned with data and the duration in the Flash crowd will be longer than the duration under the M-DoS attacks. In our analysis of CAIDA Datasets, we found all the duration of flow entries with M-DoS attacks is less than or equal to 1 second. And the duration of flow entries in the Flash crowd scene ranges from 4 to 10 seconds. Therefore, we make PFSD in the flow table to distinguish the two sides, which is defined as follows:

$$PFSD = \frac{\sum_{i}^{FlowSum} Flow_i(duration < Threshold)}{FlowSum} \quad (6)$$

where *duration* record the duration time of each flow entry. *Threshold* is used to select the flow entry under the attacks. In our experiment, it is set as 5 seconds. $Flow_i$ represents the number of flow entries after threshold filtering, i.e. the number of flow entries that are considered to be installed in the flow table by M-DoS attack, which divide the sum of flow entries to acquire the PFSD. When the network suffers M-DoS attacks, a large amount of flow entries with short time duration pour into the switch, the value of PFSD will keep pretty high, but it will still low in the Flash crowd scene.

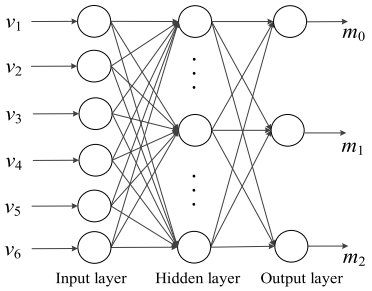


FIGURE 3. The model of BP neural network.

C. CLASSIFICATION BY FEATURES

According to the six features mentioned above, the flow entries are divided into M-DoS attacks, S-DoS attacks and normal traffic by classification algorithm after training. There are many species of classification algorithms, Naive Bayesian classification is efficient but the algorithm is simple and easy to lose data. SVM (Support Vector Machine) also has the same question. KNN (K-Nearest Neighbor) is effective to lower the cost of training but its calculation is bigger, which need to remove the little effect on classification of samples in advance. For most of the data, the classification effect of Random Forest is better. But the calculation speed is slower than a single decision tree, and it is easy to produce overfitting. The BP neural network has strong learning ability, high classification accuracy and distributed data processing ability, as well as good robustness [29], [30]. Comparing the precision rate, accuracy, F_1 score and recall rate of the above five classification algorithms in the experiment section, we found that the BP neural network has optimal performance, so a three layers BP neural network is used to establish our attack classifier. The model of classifier is shown in Figure.3.

In Figure.3, there are six input layer neurons corresponding to the proposed six feature vectors (v_1, v_2, v_3, v_4, v_5, v_6) while the number of output layer neurons is three. The three target outputs (m_0, m_1, m_2) correspond to normal traffic, M-DoS attack traffic and S-DoS attack traffic respectively.

Experiments show that when the output layer neurons use sigmoid activation function, the detection accuracy is better than use Ramp function and Threshold function. Thus, sigmoid function is selected as the activation function of our BP classifier model. In addition, we use Levenberg-Marquardt algorithm to adjust the weight and minimize the sum of squared errors E between the expected output values and actual output values as rapidly as possible [31]. The following formula is used to adjust the number of neurons in the hidden layer:

$$h = \sqrt{q + t} + a, \quad a \in [1, 10] \quad (7)$$

where h, q and t respectively denote the number of hidden neurons, input layer and the output layer [32]. The whole DoS attack detection algorithm is shown as follows:

Algorithm 2 Detection DoS Attack Algorithm

Input: Flow table $ft_1, ft_2, \ldots ft_n$
For ft_i in ft_n and $i = 1$ to n **do**
  For fr_s in ft_i and $s = 1$ to $length(ft_i)$ **do**
    $IPs = nw\_src, P_s = n\_packet, B_s = n\_byte, fe_s =$ (*IPsrc, IPdst, Portsrc, Portdst, Protocol*)
    If $n\_packet < Threshold$ **then**
      $f_p = f_p + 1$
    End if
    If $n\_byte < Threshold$ **then**
      $f_d = f_d + 1$
    End if
  End for
  Calculate the entropy (H),
  $fc_i = \{fe_1, fe_2, \ldots, fe_n\}$, $SFT_i = F_{SFT}(fc_i, fc_{i+1})$,
  $MP_i = \max(P_1, P_2, \ldots P_n)$, $MB_i = \max(B_1, B_2, \ldots B_n)$,
  $GRMMPi = \frac{|MP_{i+1} - MP_i|}{\Delta t}$, $GRMMBi = \frac{|MB_{i+1} - MB_i|}{\Delta t}$,
  $PFSP_i = f_p$ / $length(ft_i)$, $PFSD_i = f_d$ / $length(ft_i)$
  Send(H_i, SFT_i, $GRMMP_i$, $GRMMB_i$, $PFSD_i$, $PFSP_i$) to BP classifier
End for
Classifier **output** S-DoS, M-DoS or normal traffic

In algorithm 2, we collect flow table ft per second for n seconds. Every second of the flow table is composed of s flow rules fr. For each flow rule fr_s, we extract the source address nw_src, the number of matched packets n_packet and the number of matched bytes n_byte. We let (IP_{src}, IP_{dst}, $Port_{src}$, $Port_{dst}$, $Protocol$) denote a flow entry fe_s, record the number of flows with small number of packets and the number of flows with short time duration in the flow table at the same time. After traversal all s flow rules, we calculate the entropy of the source IP address H_i, use fc_i consisted of all fe to acquire the SFT_i, compute $GRMMP_i$, $GRMMB_i$, $PFSD_i$ and $PFSP_i$ of the i-th second flow table ft_i. The six-tuple (H_i, SFT_i, $GRMMP_i$, $GRMMB_i$, $PFSD_i$, $PFSP_i$) we got represent the i-th second flow table, which will be send to the trained BP classification model. The classifier then output the result with S-DoS, M-DoS or normal traffic, finally, we got the detection consequence for n seconds.

IV. EXPERIMENTS AND RESULTS ANALYSIS

In terms of experimental environment, most of the researchers used Mininet as the experimental environment in the past few years, which is standard network work emulator for SDN [29]. But software simulation is often ideal, and the test-bed composed of real hardware is a more reliable experimental environment. With the development of experimental platforms, researchers use test-bed made of hardware to conduct experiments [6], [18]. In this section, we verified our method through test-bed experiments and evaluated its performance by comparing with other methods.

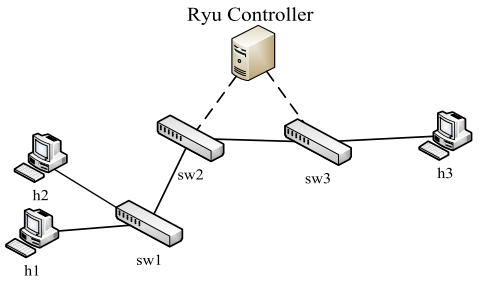


FIGURE 4. Experimental topology.

A. THE EXPERIMENT ENVIRONMENT

The experiment topology is shown in Figure.4. The testbed contains three hardware SDN switches (Centec V350), a popular Ryu controller, and three physical hosts. The controller is deployed on a server with a quad-core Intel Xeon CPU E5504 and 64GB RAM. Each host has a quad-core Intel i3 CPU and 4GB RAM. All hosts run Ubuntu 18.04.3 LTS. The OpenFlow protocol is configured as the southern interface between the controller and switches. Although the topology in Figure.4 is simple, we replayed the CAIDA dataset collected from the real network to test the proposed method. The CAIDA dataset is one of the most credible datasets. It collects various types of actual network traffic in different locations around the world, including Web, FTP, Ping, etc [33]. This dataset is widely used to simulate an actual large-scale network scenario. It is a common way that we conduct the experiment through simple hardware topology and dataset [6], [18], [19], [29].

In the test-bed, sw1 is regarded as the victim target that can store up to 1500 flow rules, and the link between sw1 and sw2 is the target shared link. h1 is configured as the attacker that launch M-DoS attacks and S-DoS attacks. The M-DoS traffic is sent to the target sw1. The S-DoS traffic is sent to h3 via the target shared link. h2 is used to generate the normal background traffic passing through sw1, sw2 and sw3 to h3.

The background traffic is composed of two public real-time datasets: *CAIDA Datasets. Anonymized Internet Traces 2016* and *FIFA world cup 1998 dataset*. *CAIDA Datasets. Anonymized Internet Traces* is a reliable dataset, which collected a variety of real network traffic types in 2016 [33]. In our background traffic, TCP flows account for 80%, ICMP flows account for 15% and UDP account for 5%. The *FIFA world cup 1998 dataset* includes all requests for the 1998 World Cup website from April 30, 1998 to July 26, 1998 [34]. During these 88 days, the World Cup venue received 1,352,804,107 requests. The *FIFA world cup 1998 dataset* is used to generate Flash crowd traffic. We use *CICIDS2017 dataset* and *HogzillaIDS dataset* as M-DoS attack traffic. The *CICIDS2017 dataset* contains approximately five days of anonymized traffic traces from July 2 to July 7, 2017. The attacks included the violent FTP, DoS, Heartbleed, Web attack, and DDoS [35]. *HogzillaIDS dataset* is an open source Intrusion Detection System (IDS) supported by SFlows, Snort, Apache Spark, libnDPI, GrayLog and HBase, which provides Network Anomaly Detection [36].

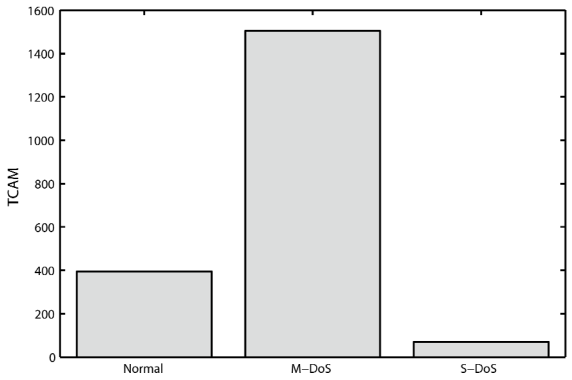


FIGURE 5. Effect of the attacks.

The protocols of M-DoS attacks are TCP and ICMP protocols, where a large proportion of 92% are ICMP packet and TCP packet account for 7% of the proportion. We use the classical LDoS attack tool developed by Rice University to send S-DoS attack flows to h3 [37]. We set the attack burst rate equaling to the link bandwidth.

B. ATTACK EFFECT

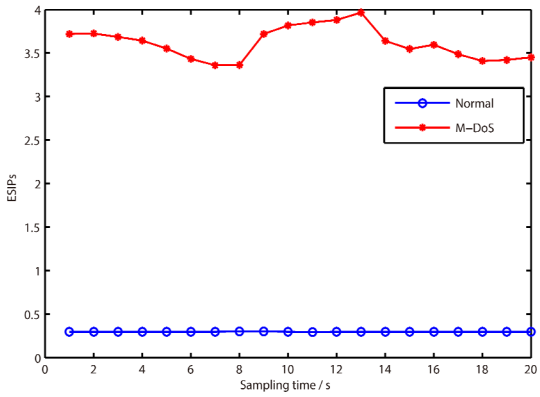
We launch M-DoS and S-DoS against target switch sw1, and observe the effect of attack by the number of flow rules in sw1, that is, TCAM resources. The TCAM of sw1 during the attack is shown in Figure.5.

From Figure.5, we can see that the TCAM value of sw1 is normally around 400. When the M-DoS attack is launched, a large number of unique streams flooded into sw1, causing the switch to install a large number of flow rules, and the TCAM resources were fully occupied, remaining at 1500. It was not possible to continue installing flow rules for subsequent new streams. During the S-DoS attack, a well-designed data stream flows through the target switch which the pulse rate just above the link bandwidth, causing the link and target port buffer of switch filled. The TCP flow in the background flow cannot be sent because of the congestion control mechanism. When S-DoS packets are idle, only a small number of UDP and ICMP streams can be sent, and the TCAM value remains around 20.

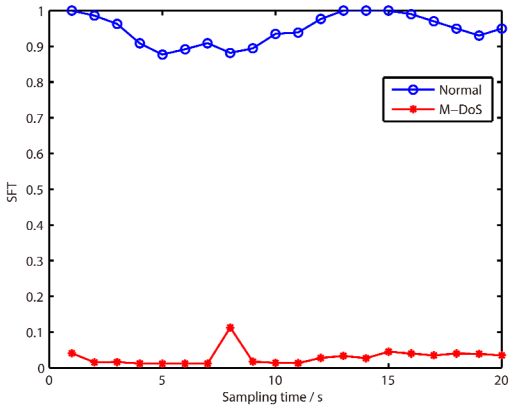
C. FEATURES ANALYSIS

We collect flow tables per second and extract the proposed features. Test results are shown in Figure.6 and Figure.7, where the curve 'M-DoS' represents the normal traffic mixed with M-DoS, and the curve 'S-DoS' represents the normal traffic mixed with S-DoS, while the normal traffic does not include the Flash crowd scene.

Twenty test results of ESIPs and SFT are shown in Figure.6. In Figure.6(a), we can see that the ESIPs of normal traffic stays around 0.35, while the ESIPs of M-DoS ranges from 3.3 to 4. In Figure.6(b), we can see that the SFT of normal traffic ranges from 0.85 to 1. However, the SFT of M-DoS traffic ranges from 0.05 to 0.15.



(a) ESIPs



(b) SFT

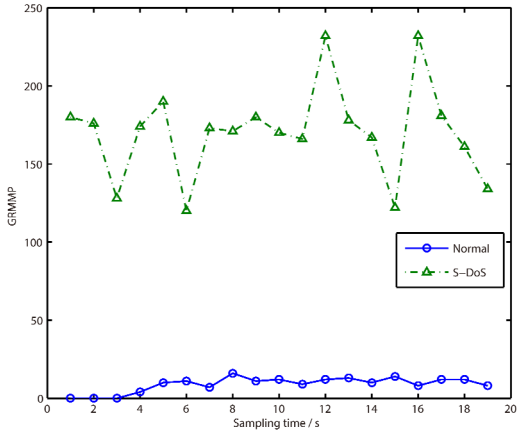
FIGURE 6. Features for M-DoS.

Figure.6 demonstrates that the ESIPs and the SFT can be used to identify M-DoS effectively.

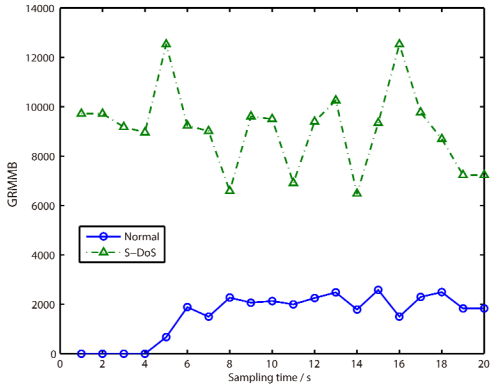
Next, we set $\Delta t = 1$s to extract the GRMMP and GRMMB. Test results are shown in Figure.7. In Figure.7(a), the value of GRMMP under normal condition are very low. However, the value of GRMMP under S-DoS is very high, which ranges from about 110 to 240. In Figure.7(b), the value of GRMMB under normal condition has some increase, but all of the value is under 2500, while the value of GRMMB under S-DoS ranges 6000 to 13000. Figure.7 indicates that we can use GRMMP and GRMMB to distinguish S-DoS accurately.

Although the above four features can accurately distinguish M-DoS attacks and S-DoS attacks, in Flash crowd scene, short-term burst traffic is easily regarded as attack flows by features for M-DoS, causing the false positive probability too high. Therefore, PFSP and PFSD are proposed to distinguish M-DoS and Flash crowd. Test result of the two features are shown in Figure.8.

We use *FIFA world cup 1998 dataset* to generate Flash crowd traffic in Figure.8, in which about 200 active flows quickly access the switch in a short time and wait for a reply, resulting in a large number of flow rules installed. In Flash crowd, most of the traffic size is less than 10KB and the intervals time of flows conforms to heavy-tailed distributions. As shown in the Figure.8(a), the value of PFSP under

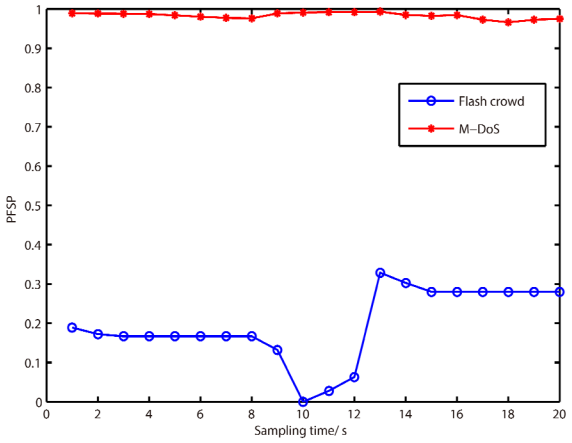


(a)GRMMP

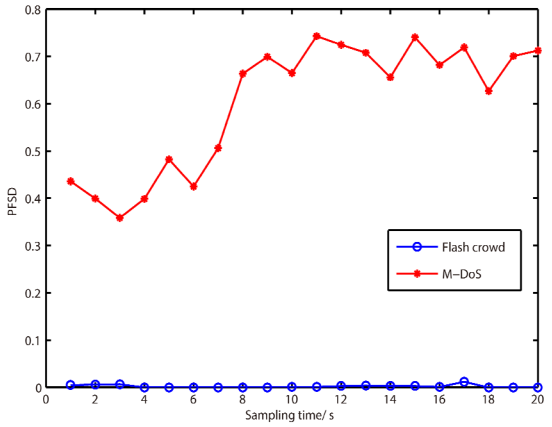


(b)GRMMB

FIGURE 7. Features for S-DoS.



(a)PFSP



(b) PFSD

FIGURE 8. Features for Flash crowd.

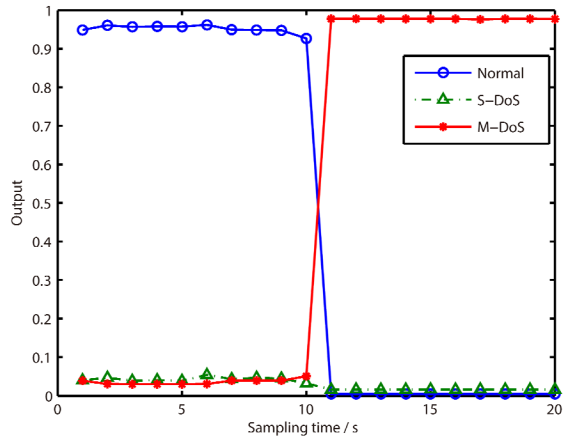
Flash crowd which is slight change range from 0 to 0.35. However, the value of PFSP under M-DoS stays about 0.98. In Figure.8(b), the value of PFSD under Flash crowd scene stay with 0 and basically unchanged, while the value of PFSD under M-DoS keep rising and goes from 0.35 to 0.75. From Figure.8, we can find that PFSP and PFSD can distinguish Flash crowd and M-DoS accurately, reducing the false positive probability.

In summary, the six features can effectively detect M-DoS and S-DoS, which can also accurately distinguish Flash crowd and DoS attack in the same time.

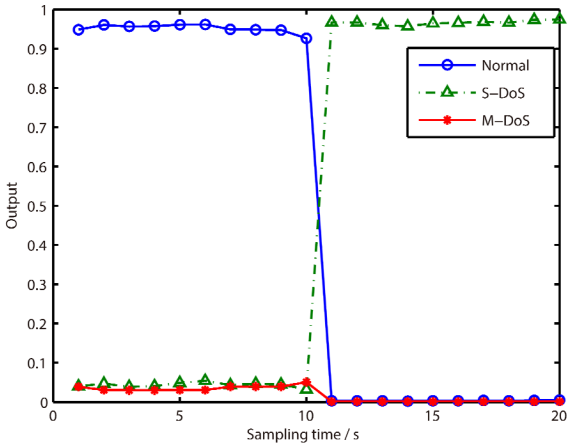
D. DETECTION RESULTS ANALYSIS

Next, we train and test our method, three distinct attack schemes are used to extensively evaluate the detection performance. Scheme 1: normal background traffic mixed with M-DoS attacks. Scheme 2: normal background traffic mixed with S-DoS attacks. Scheme 3: normal background traffic mixed with both M-DoS and S-DoS attacks. All normal traffics include Flash crowd scene. For each scheme, we choose 1000 groups of features as training sets and 400 groups of features as test sets. The maximum iteration number, the number of learning rate and the training target are respectively determined as 10000, 0.001 and 0.0001. For evaluating the our method performance, we use the trained BP classifier to classify the three scheme mentioned above. For each scheme, we collect the classifier output per second for a total of 20 seconds. The final outputs of the classifier are shown in Figure.9.

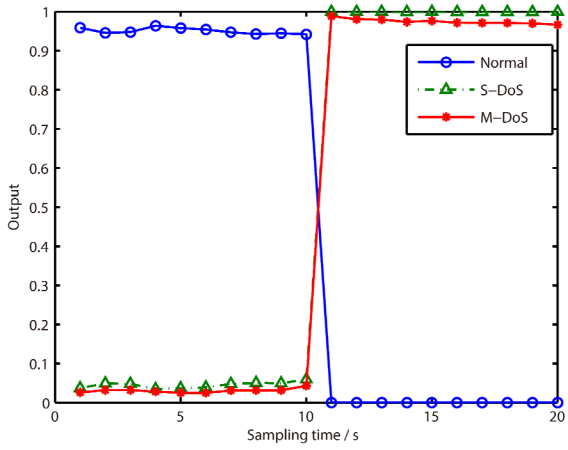
As shown in Figure.9(a), classifier shows the output is normal for the first ten seconds and the output is M-DoS after 10 seconds, consistent with scheme 1. In Figure.9(b), the classifier shows the output is normal for the first ten seconds and the classifier detects S-DoS attacks after ten seconds, consistent with scheme 2. In Figure.9(c), the output of classifier is normal in the first ten seconds, and the classifier detects both M-DoS attacks and S-DoS attacks in the last ten seconds, which is consistent with scheme 3. In Figure.9, the results of scheme 1, 2 and 3 show that the proposed approach can detection M-DoS and S-DoS accurately. According to the theory of BP classifier, normal traffic, M-DoS attack traffic and S-DoS attack traffic are given the binary output with (1, 0, 0), (0, 1, 0) and (0, 0, 1). Also, if the normal traffic is mixed with both M-DoS attacks and S-DoS attacks, the desired outputs is (0, 1, 1). The correspondence between the classifier outputs and the classification results can be expressed as Table 2.



(a) Scheme 1



(b) Scheme 2



(c) Scheme 3

FIGURE 9. Outputs of the classifier.

In order to compare the false positive rate vs. true positive rate, the Receiver Operating Characteristic curve (ROC) is obtained by adjusting the number of hidden layers [38]. As shown in Figure.10, the upper right corner is the best critical point for the classifier when the hidden layer of BP classifier is 12. Also, the Mean Square Error (MSE) is the smallest when the hidden layer neuron is 12.

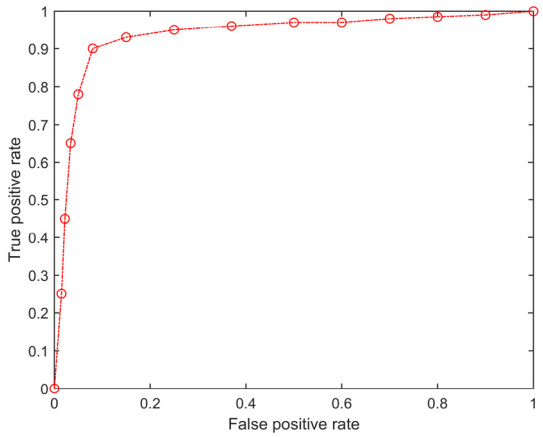


FIGURE 10. The ROC curve.

TABLE 2. Classification results.

Output	Classification result
If it is closer to (1, 0, 0)	Normal traffic
If it is closer to (0, 1, 0)	M-DoS
If it is closer to (0, 0, 1)	S-DoS
If it is closer to (0, 1, 1)	Both M-DoS and S-DoS

TABLE 3. Detection comparison of Multi-features and single-feature.

Feature types	P_D for M-DoS	P_D for S-DoS	P_{FP} for M-DoS in Flash crowd
ESIPs	90.1%	NA	17.4%
SFT	92.4%	NA	19.5%
GRMMP	NA	92.6%	NA
GRMMB	NA	92.9%	NA
PFSD	88.7%	NA	2.4%
PFSP	89.9%	NA	2.5%
Multi-features	98.9%	99.2%	2.3%

Next, we put six single features and multiple features into BP classification model with same condition (Set training number to 1000 and hidden layer neuron to 12), The test results for the same test set are shown in table 3, including accurate detection probability (P_D) and the false positive probability (P_{FP}).

From Table 3, we can see that ESIPs and SFT can detect M-DoS attack, but unable to detect S-DoS attack and the false positive probability is too high. GRMMP and GRMMB can only detect the S-DoS attack. Although PFSD and PFSP can distinguish M-DoS and Flash crowd, their accurate detection probability is not very good. The Multi-feature which combine the advantages of six features has high accurate detection probability and low false positive probability, is a comprehensive performance DoS detection method.

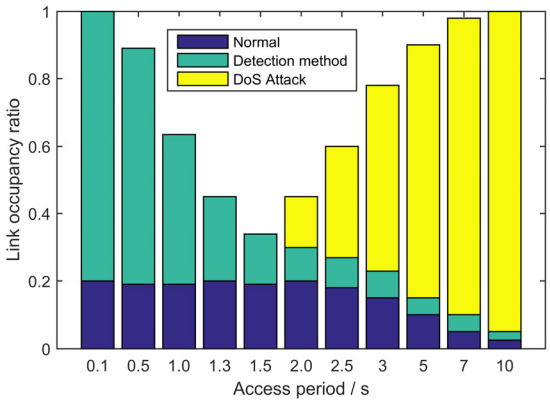


FIGURE 11. Link occupancy raito.

Our detection method needs to use the controller to query the flow table information of the switch. The flow table information will be transmitted back through the connection link. However, too frequent requests will dramatically consume shared-link and cause link congestion. In order to avoid the congested link problem and ensure the real-time detection, we need to select the appropriate access period of information collection module. Therefore, we tested link occupancy ratio between the controller and the switch under different access periods. The experimental results are shown in Figure.11.

As shown in Figure.11, we tested the link occupancy in different states. We can see that the link occupancy ratio is about 0.2 under normal state. Then we deploy detection algorithm and DoS attack. When the access period is less than 1s, frequent requests for information cause the link between the controller and the switch to be congested or even paralyzed (the link ratio reaches 1). As the access period gradually increases, the additional link consumption due to the detection method gradually decreases. But when the access period grows to 2s, the detect delay caused by the long period makes some attacks consume link resources. This consumption continues to increase with the increase of the access period, and the attack fills the entire link when the period reaches 10s. Through experiments, we found that when the access period is 1.5s, the detection method can maintain a low consumption of the link while ensuring the real-time detection.

E. BP MODEL ANALYSIS

We compare and analyze BP model constructed in this paper with other classification models. All classification models use the same features data set as the input of the model.

Experiment through four evaluation indexes consisting of precision (P), accuracy (Acc), F_1 score (F_1) and recall rate (R) to evaluate the detectability of the model, which can accurately represent the performance of the model. Where, TP (true positive) is the number of samples that are correctly judged by the classification model with the actual DoS attack type; TN (true negative) is the number of samples that are correctly judged by the classification model with the actual normal type; FN (false negative) is the number of samples which actual type is DoS attack but is misclassified as normal type by the classification model; FP (false positive) is the number of samples which actual type is normal but is misclassified as DoS attack type by the classification model.

Precision (P) indicates the percentage of packets actually attacked in the data packet judged by the model as the attack type, which is calculated as follow:

$$P = \frac{TP}{FP + TP} \tag{8}$$

Accuracy (Acc) indicates the percentage of packets that the model judges to be correct in the total number of data groups, which is calculated as follow:

$$Acc = \frac{TN + TP}{FN + TN + TP + FP} \tag{9}$$

F_1 score (F_1) represents the harmonic average value of accuracy and recall rate, which can more accurately evaluate model performance. Formula is shown as follow:

$$F_1 = \frac{2PR}{P + R} \tag{10}$$

The recall rate (R) indicates the percentage of packets that the model judges as the attack type in the number of all attack packets, which is calculated as follow:

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

Next, we train and test the five classifiers (SVM, BP, Naïve Bayesian, Random Forest and KNN) by classifying the three scheme mentioned above. Meanwhile, we compare different values of the same classifier parameter for each classifier, in order to adjust each classifier to the optimal state. We found that the SVM classifier works best when the number of support vectors is 27 and the number of support vectors in each category is 9. The detection rate is the highest and the MSE (mean square error) is the smallest in the BP classification model when the hidden layers is 12. Compared with the common model in Naïve Bayesian classifier, polynomial model is better. When the maximum depth is 50, the classification effect of random forest is the best. And in KNN, when we compare the Manhattan distance, the Euclidean distance, the hamming distance and the min distance, we find that the Euclidean distance is the most accurate way to calculate the distance between the attacks and the normal traffic. We adjusted the parameters of the five classifiers to the optimal, used the same features data set of flow table for training and testing, and calculated their Acc, P, R and F_1 score. The experimental test result is shown in Table4.

As shown in Table 4, we compare accuracy, precision rate, recall rate and F_1 score of different classification model with the same input. We found that the BP neural network has optimal performance, which is more suitable to our study. And the performance of BP classifier is optimal when the hidden layers is 12.

TABLE 4. Detection comparison of different classification models.

Model name	*Acc*	*P*	*R*	F_1
SVM	96.72%	96.47%	95.84%	96.77%
BP	98.9%	98.17%	97.94%	98.12%
Naïve Bayesian	94.44%	97.12%	97.87%	97.53%
Random Forest	98.03%	97.98%	97.66%	97.11%
KNN	97.47%	95.82%	93.74%	93.55%

TABLE 5. Detection performance of different methods.

Approach	M-DoS	S-DoS	Both	Detection time(s)
	$P_D/P_{FN}/P_{FP}$(%)	$P_D/P_{FN}/P_{FP}$(%)	$P_D/P_{FN}/P_{FP}$(%)	
Naïve Bayesian	95.4/8.5/6.2	NA	95.4/8.5/6.2 for M-DoS	0.91
generalized entropy	96.3/5.2/6.6	NA	96.3/5.2/6.6 for M-DoS	1.10
SGuard	96.9/6.9/7.8	NA	96.9/6.9/7.8 for M-DoS	1.89
Random Forest	97.3/0.9/3.3	NA	97.3/0.9/3.3 for M-DoS	3.22
Our approach	98.9/0.5/2.4	99.2/0.7/2.7	98.9/0.9/2.9	1.11

F. COMPARED ANALYSIS

Then, we prove our strengths by comparing with other methods. We test the false positive probability (P_{FP}), the accurate detection probability (P_D) and the false negative probability (P_{FN}) of our approach under different attack schemes. Also, we implement other four existing approaches in our experiment environment, including machine learning methods and traditional methods [27]–[30]. We compare these methods with our approach in the same experiment condition. Performance comparison is shown in Table 5.

Table 5 indicates that Naïve Bayesian, generalized entropy, SGuard and Random Forest are available for detecting M-DoS, but can not detect S-DoS. However, our approach can detect both M-DoS and S-DoS. Moreover, our approach has higher P_D, lower P_{FN} and P_{FP} than the other methods. The reason mainly lies in the face that we designed two features to reduce the false positive probability in Flash crowd scene, which are not considered by other methods. Therefore, in the same Flash crowd scene, our performance is better.

The detection time in Table 5 is defined as the time from sample collection to classification, excluding offline training time. We can see that our approach costs less time than SGuard and Random Forest. This is because that, the Random Forest has more hidden layers, SGuard needs to preprocess the data and the implementation of SOM requires a 40×40 matrix of neurons, So the overhead is higher during detection. And the Naïve Bayesian and generalized entropy take less time due to their small computation, but small computation may lead to low detection rate. To sum up, the proposed features not only consider the variation of a single indicator in the flow table, but also globally consider the similarity of flow entries over time. Therefore, these features have better resolution, i.e. they are not easily confused with normal flow.

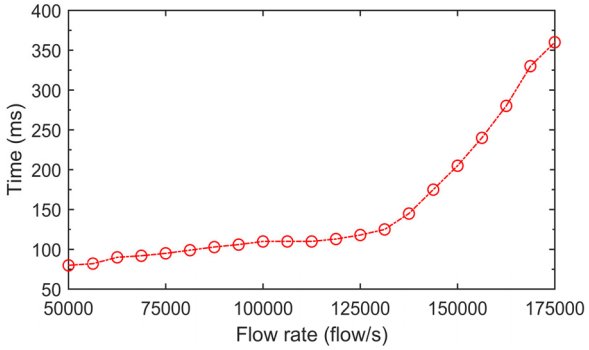


FIGURE 12. Execution time on the controller.

Finally, we explore how the execution time of our method on the controller varies with the network scale. Literature [39] indicated that one controller commonly could support 20 to 70 switches, and the average flow rate of each switch was 2500 flow/s. Accordingly, in our experiment environment, we explored how the execution time varies with network scale by changing the number of switches from 20 to 70. The flow rate of each switch is 2500 flow/s, that is, the flow rate through controller is 50000 flow/s to 175000 flow/s [39]. Figure.12 presents the execution time as a function of the flow rate.

Figure.12 indicates that the execution time of our approach slowly increases with the growth of the flow rate when the flow rate is less than 125000 flow/s (50 switches), and gradually stabilizes at 110ms. As the network further scales up, the controller itself cannot satisfy the all requirements of the network traffic, so the execution time rapidly increases with the growth of flow rate, and eventually up to 360ms when the flow rate is 175000 flow/s (70 switches). Test results show that our approach is efficient and lightweight.

V. CONCLUSION

In SDN, DoS attacks on data plane have attracted wide attention. Through research, we discovered two main DoS attacks (M-DoS attack and S-DoS attack) against SDN data plane. Flow table provides important information for detecting these two DoS attacks. This paper extracts six new features from SDN flow table for DoS attack detection, and designs a BP neural network to detect M-DoS attack and S-DoS attack, which can distinguish DoS and Flash crowd in the same time. The proposed approach is evaluated in test-bed experiment. The test results show that, compared with other methods, our method improves the detection probability by 1% to 3%, while the false positives probability decreases by 5% to 8%. In addition, the security of control plane or application plane is also very important, so we will focus on the security of these two planes in the future. We will explore to combine the dataplane switches with intelligent logic or applications to detect attacks to its resources (Flow Table/TCAM), and select the appropriate attributes of application layer protocol

to calculate the joint entropies. We will work to combine multiple planes to detect and defend against DoS attacks, while trying to balance the resource consumption of each plane in the detection.

REFERENCES

[1] R. Amin, M. Reisslein, and N. Shah, "Hybrid SDN networks: A survey of existing approaches," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 3259–3306, 4th Quart., 2018.

[2] H. Park, B. Cho, I. Hwang, and J. R. Lee, "Study on the SDN-IP–based solution of well-known bottleneck problems in private sector of national R&E network for big data transfer," *Concurrency Comput., Pract. Exper.*, vol. 30, no. 1, p. e4365, 2018.

[3] Y. Wang, H. Chen, X. Wu, and L. Shu, "An energy-efficient SDN based sleep scheduling algorithm for WSNs," *J. Netw. Comput. Appl.*, vol. 59, pp. 39–45, Jan. 2016.

[4] T. Li, J. Chen, and H. Fu, "Application Scenarios based on SDN: An Overview," in *Proc. J. Phys., Conf. Apr.*, 2019, pp. 1–9.

[5] D. Gao, Z. Liu, Y. Liu, C. H. Foh, T. Zhi, and H.-C. Chao, "Defending against packet-in messages flooding attack under SDN context," *Soft Comput.*, vol. 22, no. 20, pp. 6797–6809, Oct. 2018.

[6] T. V. Phan and P. Minho, "Efficient distributed denial-of-service attack defense in SDN-based cloud," *IEEE Access*, vol. 7, pp. 18701–18714, 2019.

[7] R. Xie, M. Xu, J. Cao, and Q. Li, "SoftGuard: Defend against the low-rate TCP attack in SDN," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[8] R. Neres Carvalho, J. Luiz Bordim, and E. Adilio Pelinson Alchieri, "Entropy-based DoS attack identification in SDN," in *Proc. IEEE Int. Parallel Distrib. Process. Symp. Workshops (IPDPSW)*, May 2019, pp. 627–634.

[9] M. H. H. Khairi, S. H. S. Ariffin, N. M. A. Latiff, and A. S. B. Abdullah, "A review of anomaly detection techniques and distributed denial of service (DDoS) on software defined network (SDN)," *Eng. Technol. Appl. Sci. Res.*, vol. 8, no. 2, pp. 2724–2730, 2018.

[10] T. S. Huang, P. Y. Hsiung, and B. C. Cheng, "Mitigating DoS attacks in SDN using offloading path strategies," *J. Internet Technol.*, vol. 20, no. 4, pp. 1281–1285, 2019.

[11] S. Gao, Z. Li, B. Xiao, and G. Y. Wei, "Security threats in the data plane of software-defined networks," *IEEE Network*, vol. 32, no. 4, pp. 108–113, Jul./Aug. 2018.

[12] S. Saharan and V. Gupta, "Prevention and mitigation of DNS based DDoS attacks in SDN environment," in *Proc. 11th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2019, pp. 571–573.

[13] S. Li, Y. Cui, Y. Ni, and L. Yan, "An effective SDN controller scheduling method to defence DDoS attacks," *Chin. J. Electron.*, vol. 28, no. 2, pp. 404–407, Mar. 2019.

[14] Open Networking Foundation. *Openflow Switch Specification (v1.3.0), Version 1.3.0 (Wire Protocol 0×04)*. Accessed: 2012. [Online]. Available: https://www.opennetworking.org/

[15] C. Zhang, G. Hu, G. Chen, A. K. Sangaiah, P. Zhang, X. Yan, and W. Jiang, "Towards a SDN-based integrated architecture for mitigating IP spoofing attack," *IEEE Access*, vol. 6, pp. 22764–22777, 2018.

[16] P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.

[17] J. Cao, M. Xu, Q. Li, and J. Zheng, "Disrupting sdn via the data plane: A low-rate flow table overflow attack," in *Proc. 13th Int. Conf. Secur. Privacy Commun. Syst.*, Apr. 2017, pp. 356–376.

[18] T. A. Pascoal, Y. G. Dantas, I. E. Fonseca, and V. Nigam, "Slow TCAM exhaustion DDoS attack," in *Proc. IFIP Int. Conf. ICT Syst. Secur. Privacy Protection*, May 2017, pp. 17–31.

[19] B. Yuan, D. Zou, S. Yu, H. Jin, W. Qiang, and J. Shen, "Defending against flow table overloading attack in software-defined networks," *IEEE Trans. Services Comput.*, vol. 12, no. 2, pp. 231–246, Mar. 2019.

[20] M. Yue, Z. Wu, and J. Wang, "Detecting LDoS attack bursts based on queue distribution," *IET Inf. Secur.*, vol. 13, no. 3, pp. 285–292, May 2019.

[21] R. Nagarathna and S. Shalinie, "SLAMHHA: A supervised learning approach to mitigate host location hijacking attack on SDN controllers," in *Proc. 4th Int. Conf. Signal Process., Commun. Netw. (ICSCN)*, Mar. 2017, pp. 1–7.

[22] L. XU, J. HUANG, S. HONG, J. ZHANG, and G. GU, "Attacking the brain: Races in the SDN control plane," in *Proc. 26th USENIX Secur. Symp. USENIX Secur.*, 2017, pp. 451–468.

[23] J. H. Cao, Q. Li, R. J. Xie, K. Sun, G. Gu, M. Xu, and Y. Yang, "The crosspath attack: Disrupting the SDN control channel via shared links," in *Proc. 28th USENIX Secur. Symp. USENIX Secur.*, Aug. 2019, pp. 19–36.

[24] M. Yue, Z. Wu, and M. Wang, "A new exploration of FB-shrew attack," *IEEE Commun. Lett.*, vol. 20, no. 10, pp. 1987–1990, Oct. 2016.

[25] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: Detecting security attacks in software-defined networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2015, pp. 1–15.

[26] L.-D. Chou, C.-C. Liu, M.-S. Lai, K.-C. Chiu, H.-H. Tu, S. Su, C.-L. Lai, C.-K. Yen, and W.-H. Tsai, "Behavior anomaly detection in SDN control plane: A case study of topology discovery attacks," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 357–362.

[27] Z. P. Liu, Y. P. He, W. S. Wang, and B. Zhang, "DDoS attack detection scheme based on entropy and PSO-BP neural network in SDN," *China Commun.*, vol. 16, no. 7, pp. 144–155, Jul. 2019.

[28] K. Kirutika, V. Vetriselvi, R. Parthasarathi, and G. S. V. Rao, "Controller monitoring system in software defined networks using random forest algorithm," in *Proc. Int. Carnahan Conf. Secur. Technol. (ICCST)*, Oct. 2019, pp. 1–6.

[29] T. Wang and H. Chen, "SGuard: A lightweight SDN safe-guard architecture for DoS attacks," *China Commun.*, vol. 14, no. 6, pp. 113–125, 2017.

[30] M. Prasath and B. Perumal, "Network attack prediction by random forest : Classification method," in *Proc. 3rd Int. Conf. Electron., Commun. Aerosp. Technol. (ICECA)*, Jun. 2019.

[31] A. F. Lzmailov, M. V. Solodov, and E. I. Uskov, "A globally convergent Levenberg–Marquardt method for equality-constrained optimization," *Comput. Optim. Appl.*, vol. 72, no. 1, pp. 215–239, 2019.

[32] S. Wu and W. Cao, "Parametric model for microwave filter by using multiple hidden layer output matrix extreme learning machine," *IET Microw., Antennas Propag.*, vol. 13, no. 11, pp. 1889–1896, Sep. 2019.

[33] CAIDA Datasets. *Anonymized Internet Traces 2015*. Accessed: 2018. [Online]. Available: https://data.caida.org/datasets/passive-2015

[34] *FIFA World Cup 1998 Dataset*. Accessed: 2018. [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[35] *CIC Datasets. IDS-2017*. Accessed: 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html

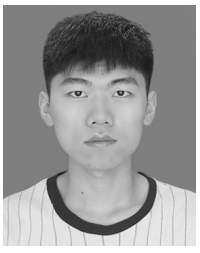
[36] *Hogzilla IDS Datasets*. Accessed: 2019. [Online]. Available: https://ids-hogzilla.org#content

[37] E. W. Knightly and A. Kuzmanovic. (2004). *Shrews: Low-Rate TCP-Targeted Denial of Service Attacks [EB/OL]*. [Online]. Available: http://www.cs.northwestern.edu/~akuzma/rice/shrew/ 2004.

[38] S. S. Hassan, H. Huttunen, J. Niemi, and J. Tohka, "Bayesian receiver operating characteristic metric for linear classifiers," *Pattern Recognit. Lett.*, vol. 128, pp. 52–59, Dec. 2019.

[39] P. Song, Y. Liu, T. Liu, and D. Qian, "Controller-proxy: Scaling network management for large-scale SDN networks," *Comput. Commun.*, vol. 108, pp. 52–63, Aug. 2017.

MENG YUE received the Ph.D. degree in information and communication engineering from Tianjin University, China, in 2017. He is currently an Associate Professor with the School of Electronics and Information Engineering and Automation, Civil Aviation University of China. His current research interests include information security and cloud computing.



HUAIYUAN WANG is currently pursuing the master's degree in electronics and communications engineering with the Civil Aviation University of China. His current research interests are networks and information security in software defined networking.

LIANG LIU received the master's degree in communication and information system from the Civil Aviation University of China. He is currently working as an Assistant Experimenter with the School of Electronic Information and Automation, Civil Aviation University of China. His current research interests are networks and information security, including defense of future network security and denial of service attacks.

ZHIJUN WU received the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, China, in 2004. He is currently a Professor with the Civil Aviation University of China. He is also the Supervisor of Ph.D. degree candidates at Tianjin University, China, and a Security at the Beijing University of Posts and Telecommunications, China. His current research interests include denial-of-service attacks, and security in big data and cloud computing.