Hindawi
Security and Communication Networks
Volume 2018, Article ID 9804061, 8 pages
https://doi.org/10.1155/2018/9804061

WILEY | Hindawi

Research Article

A DDoS Attack Detection Method Based on SVM in Software Defined Network

Jin Ye,[1,2] Xiangyang Cheng,[1,2] Jian Zhu,[1,2] Luting Feng,[1,2] and Ling Song[1,2]

[1]*School of Computer and Electronic Information, Guangxi University, Nanning 530004, China*
[2]*Guangxi Key Laboratory of Multimedia Communications and Network Technology, Nanning 530004, China*

Correspondence should be addressed to Ling Song; aling7197_cn@sina.com

Received 13 October 2017; Revised 28 December 2017; Accepted 24 January 2018; Published 24 April 2018

Academic Editor: Zhiping Cai

The detection of DDoS attacks is an important topic in the field of network security. The occurrence of software defined network (SDN) (Zhang et al., 2018) brings up some novel methods to this topic in which some deep learning algorithm is adopted to model the attack behavior based on collecting from the SDN controller. However, the existing methods such as neural network algorithm are not practical enough to be applied. In this paper, the SDN environment by mininet and floodlight (Ning et al., 2014) simulation platform is constructed, 6-tuple characteristic values of the switch flow table is extracted, and then DDoS attack model is built by combining the SVM classification algorithms. The experiments show that average accuracy rate of our method is 95.24% with a small amount of flow collecting. Our work is of good value for the detection of DDoS attack in SDN.

1. Introduction

With the continuous development of network technology, the ceaseless expansion of network business needs, and rapid growth of the Internet economy in the Internet age, the services of network with important business and industry information have been spread to the production and life of current society. The emergence of DDoS attacks can lead to abnormalities in the related network services, causing huge economic losses and even causing other catastrophic consequences. DDoS attacks are one of the serious network security threats facing the Internet. It is a key research topic in the security field to detect DDoS attacks accurately and quickly. SDN is an emerging network innovation architecture that separates the network data plane and the control plane [1, 2], which has the characteristics of network programmable, centralized management control, and interface opening.

Network attackers attack network bandwidth, system resources, and application resources, to achieve the effect of denial of service attacks. DDoS attacks show the increasing scale of attack; the attack mode is more intelligent. The difficulties of DDoS attack detection are as follows: (1) the attack traffic characteristics not being easy to identify; (2) the lack of collaboration between the coherent network nodes; (3) the change of the attack tool being strengthened, with the threshold of its use decreasing; (4) the widely used address fraud making it difficult to trace the source of the attack; (5) the duration time of attack being short and response time being limited.

In the traditional network architecture, the main methods of DDoS attack detection technology can be divided into attack detection based on traffic characteristics and attack detection based on traffic anomaly. The former mainly collects all kinds of characteristics information related to the attack and establishes a characteristics database of DDoS attack. By comparing and analyzing the data information of the current network data packet and characteristics database, we can judge whether it is attacked by DDoS or not. The main implementation methods are characteristics match, model reasoning, state transition, and expert systems. The latter is mainly to establish traffic model and analysis of abnormal flow changes, to determine whether the traffic is abnormal or not, so as to detect whether the server was attacked.

Under the innovative architecture environment of SDN, deep packet analysis is available through the full network view [3, 4]. It supports quick response and update of traffic policies and rules. The SDN has the capability of perceived control of the global visualization view, flexible

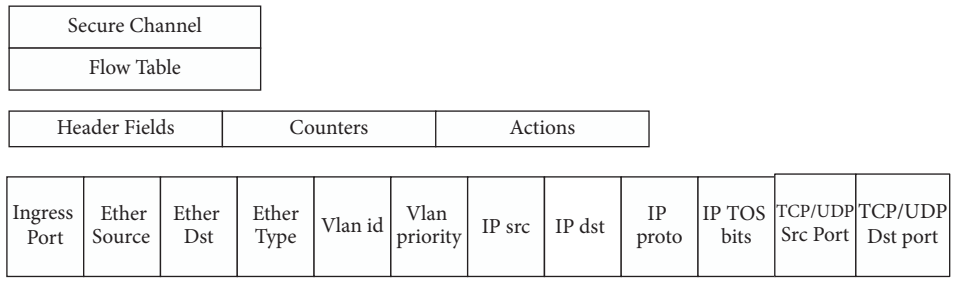


Figure 1: Flow table structure.

and schedulable rapid deployment capability, and service open intelligent scheduling capability. While ensuring network services and reducing deployment costs, the software defined network enhances the quality of user experience and facilitates the promotion of the whole network deployment.

Researchers aimed at traditional network architecture proposed a lot of DDoS attack detection methods. Lin and Wang [5] proposed a DDoS attack detection and defense mechanism based on SDN, but the method used three Openflow management tools with sFlow standard to perform anomaly detection, so the deployment and operation are complex. Yang et al. [6] dished a method in which the flow information and the IP entropy characteristic information are combined, which is detected by a single flow information and IP entropy characteristic information, which has a higher and more accurate detection effect. Although information entropy is flexible and convenient, it still needs to be combined with other technologies in determining the threshold and multielement weight distribution. Saied et al. [7] advanced that based on analysis the characteristics of each protocol of TCP/UDP/ICMP through the training ANN algorithm to detect DDoS attacks, the method needs to distinguish packet protocol, which is complex and inefficient.

In [8], the SOM algorithm is used to detect DDoS attacks by extracting the flow statistics related to DDoS attacks. This method has the characteristics of low consumption and high detection rate. The key point lies in the extraction of time interval. The disadvantage of this method is that the detection has a certain hysteresis and the attack behavior is not timely and accurately found. In [9], the authors proposed a framework for detection and mitigation of DDoS attacks in a large-scale network, but it is not suitable for small-scale deployment. In [10], a DDoS attack detection mechanism based on a legitimate source and destination IP address database is proposed. Based on the nonparametric cumulative algorithm CUSUM, it analyzes the abnormal characteristics of the source IP address and the destination IP address when the DDoS attack occurs and effectively checks the DDoS attack, but the method needs to adjust and determine the threshold.

It is concluded that DDoS attack detection in SDN networks mainly includes information entropy and utilization of data mining algorithm, in which the more popular is the SOM algorithm. Due to the high false positive rate of information entropy, the SOM algorithm needs to determine the number of neurons in advance. Therefore, in this paper, we summarize the characteristics of several DDoS attacks, then collect the switch flow table information, extract the six-tuple characteristic values matrix, and establish their SVM classification model. The algorithm can process multidimensional data and map the low-dimensional nonlinear separable data into the high-dimensional feature space to make it linearly separable and able to be classified with high accuracy. At present, the algorithm is widely used in anomaly detection and classification.

This paper is organized as follows: Section 1 describes the introduction; Section 2 gives a detailed description of the SVM classification model; Section 3 illustrates the experimental method presented in this paper; Section 4 summarizes the paper.

2. DDoS Detection Based on Support Vector Machine (SVM)

In the SDN architecture, the Openflow switch forwards the main network data at a high speed [11]. The SDN controller is responsible for the forwarding and management of the forwarding decision and the collection of traffic information of switches. In the SDN switch, the core data structure of the forwarding policy management control is the flow table [12]. The SDN manages the relevant network traffic by searching the flow table entries, where the flow entry can forward the packet to one or more interfaces. Each entry includes the header field, the counters, and the actions. The packet forwarding of the switch is based on the flow table. Each flow table is composed of multiple flow entries. The flow table entries form the rules for data forwarding. Figure 1 shows the flow table entry structure diagram.

The flow diagram of the attack detection consists mainly of the flow state collection, the extraction characteristic values, and the classifier judgment, as shown in Figure 2. The flow state collection periodically sends a flow table request to the Openflow switch and sends the flow table information replied from the switch to the flow state collection. The characteristic values extraction is mainly responsible for extracting the characteristic values related to the DDoS attack from the switch flow table and composing the six-tuple characteristic values matrix. Six-tuple characteristic values information is classified by using an SVM-based algorithm [13] to distinguish between normal traffic and attacking abnormal traffic.

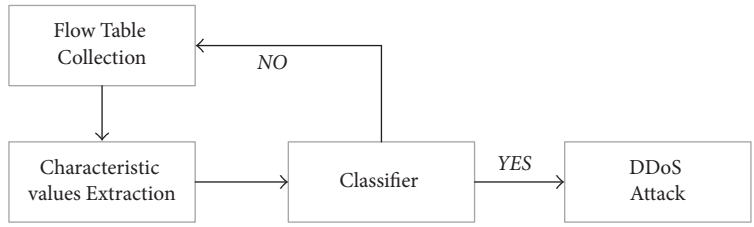


Figure 2: Attack detection process.

2.1. Flow Status Collection. In the SDN network environment, the collection of the flow table status information is mainly accomplished through the Openflow protocol. The switch responds to the *onp_flow_stats_request* message periodically sent by the controller, and the time interval between getting the flow tables should be moderate, setting the flow table obtaining period to be consistent with the flow deleting time set by the floodlight controller and running the "sudo ovs-ofct1 dump-flows s1" command to collect the status information of the flow table. The flow table information extracted by the switch is given as follows:

NXST_FLOWreply($x$id = 0×4) : cookie = 0×0, duration = 21.098 s, table = 0, n_packets = 1, n_bytes = 42, idle_timeout = 60, idle_age = 21, priority = 65535, arp, in_port = 2, vlan_tci = 0×0000, dl_src = $c6 : 76 : 11 : 0a : 4c : 78$, dl_dst = $82 : 0d : bf : d2 : ad : f0$, arp_spa = 10.0.0.3, arp_tpa = 10.0.0.1, arp_op = 1actions = output : 1.

2.2. Extract the Characteristic Values. When DDoS attack occurs on the network, for it is controlled by the program, the network will randomly forge a large number of source IP addresses to send a certain size of the packet to attack the target. In the network, the attack flow shows certain similarity, regularity, and then it can be detected by analyzing the characteristic values information of the flow table. In [14], the author does not mention the change of the speed of source port in attack detection when extracting the traffic characteristic values, and a large number of new port addresses were randomly generated in the attack process.

In this paper, some existing research on SDN is analyzed and compared and the data analysis and processing are carried out by extracting the flow status information on the basis of previous research. The following six-tuple characteristic values related to DDoS attacks are obtained for DDoS attack detection.

(1) The speed of source IP (SSIP) is the number of source IP addresses per unit of time:

$$\text{SSIP} = \frac{\text{Sum\_IP}_{\text{src}}}{T}, \tag{1}$$

where $\text{Sum\_IP}_{\text{src}}$ is the source IP number and T is the sampling interval. In the event of an attack, a large number of attacks are generated by random forgery to send data packets, the source IP address number will increase rapidly.

(2) The speed of source port (SSP) is the number of source ports per unit of time

$$\text{SSP} = \frac{\text{Sum\_port}_{\text{src}}}{T}, \tag{2}$$

where $\text{Sum\_port}_{\text{src}}$ is the number of attack source ports. When a large number of attack requests occur, a large number of port numbers are randomly generated.

(3) The Standard Deviation of Flow Packets (SDFP), that is, the standard deviation of the number of packets in the T period, is as follows:

$$\text{SDFP} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\text{packets}_i - \text{Mean\_packets}\right)^2}, \tag{3}$$

where $\text{Mean\_packets} = (1/N)\sum_{i=1}^{N}\text{packets}_i$ represent the average number of the packets in the T period. N is the total number of flow entries per period, in the event of an attack; in order to produce the attack effect, the general attack data packets are relatively small and the standard deviation of flow packets will be smaller than the normal flow.

(4) The Deviation of Flow Bytes (SDFB), that is, the standard deviation of the number of bits in the T period, is as follows:

$$\text{SDFB} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\text{bytes}_i - \text{Mean\_bytes}\right)^2}, \tag{4}$$

where $\text{Mean\_bytes} = (1/N)\sum_{i=1}^{N}\text{bytes}_i$, represent the average of the number of bits in the T period. In the event of an attack, in order to reduce the packet load, attacker will send a smaller bit of data packets and the standard deviation flow bits will be smaller than the normal flow.

(5) The speed of flow entries (SFE), that is, the number of flow entries per unit time, is as follows:

$$\text{SFE} = \frac{N}{T}. \tag{5}$$

In the event of an attack, the number of flow entries per unit time increases dramatically, significantly higher than the normal value.

(6) The Ratio of Pair-Flow (RPF), that is, the ratio of interactive flow entries to total flow entries, is as follows:

$$\text{RPF} = \frac{2 * \text{Pair\_Sum}}{N}, \tag{6}$$

where Pair_Sum is the number of interactive flow entries. Under normal circumstances, the source host sends a request to the destination host to generate an interactive flow, which constitutes the following conditions.

The source IP of packet_i is the same as the destination IP of packet_j. The destination port number of packet_i is the same as the source port number of packet_j. The destination IP of packet_i is the same as the source IP of packet_j, and the source port number of packet_i is the same as the destination port number of packet_j. There will be two interactive flow entries in the flow table that satisfy Formula (7)

$$\begin{aligned} \text{Src\_IP}_i &= \text{Dst\_IP}_j, \\ \text{Src\_port}_i &= \text{Dst\_port}_j, \\ \text{Src\_IP}_j &= \text{Dst\_IP}_i, \\ \text{Dst\_port}_j &= \text{Src\_port}_i. \end{aligned} \tag{7}$$

When an attack occurs, the flow entries sent to the destination host in a T period increase sharply, the destination host cannot respond to the interactive flow in time, and in genera the attacker typically uses massive pseudosource addresses when attacking, so the number of interactive flow entries per will drop in the T period.

2.3. Classifier Judgment. We can think of attack detection as a classification problem, that is, classifying the given data and judging that whether the current network state is normal or abnormal. In the classifier judgment, the extracted six-tuple characteristic values are used for classification learning to determine whether the traffic is abnormal. Attack detection of the basic process is as follows: the network data is extracted as a six-tuple characteristic values sequence according to the time interval, and the sample sequence is given a {normal, abnormal} flag, which represents the two states of the network.

The appropriate machine learning algorithm is selected to construct the detection model according to the sequence of characteristic values samples and the unlabeled characteristic values samples are classified by using the model. This paper chooses a classification learning method based on support vector machine (SVM) algorithm [13, 15]. SVM is a learning method based on statistical learning theory. It can get good classification results without a lot of training data. It maps the nonlinearly separable sample set to a high-dimensional or even infinite dimensional feature space to make it linearly separable and find the optimal classification surface in this high-dimensional feature space. The kernel function in SVM effectively solves the problem of dimensionality disaster caused by high-dimensional mappings and enhances the ability of processing high dimension small sample data.

SVM is applied to DDoS attack detection with good accuracy. The DDoS attack detection method proposed in this paper uses a supervised learning algorithm. Firstly, flow table entries in the switch are sampled at a time interval T, and the characteristic values of the flow table entries in each sampling are calculated to obtain a sample set Z, which is expressed as $Z = (X, Y)$, where X represents flow table entries six-tuple characteristic values matrix, Y is the category marker vector corresponding to X: "0" represents normal state, and "1" represents attacked state. In the experiment, we attacked during $T20$–$T40$ periods. We marked the corresponding class labelled "1," and the remaining class labels were all "0" and then used the SVM classifier to train the sample set to obtain its parameters. Finally, we use trained SVM model to classify the unlabeled samples. If there is a sample marked "1," it is considered that an attack was made during the corresponding detection period.

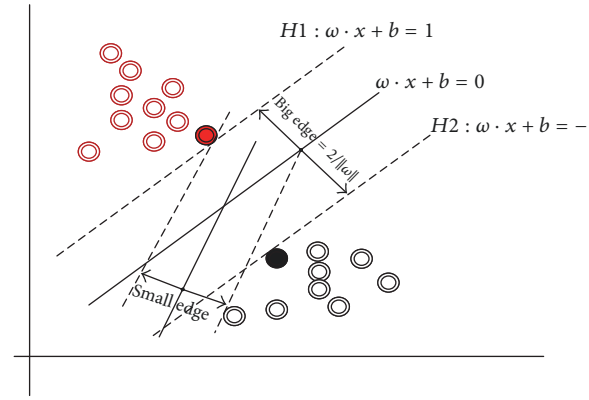


Figure 3: Classification hyperplane.

2.4. SVM. SVM is derived from the linearly separable optimal classification hyperplane, and its basic idea can be explained by the two-dimensional case of Figure 3. There is a training set $D = \{(X_1, y_1), (X_2, y_2), \ldots, (X_n, y_n)\}$, where X_i is the characteristic vector of the training sample and y_i is the associated class label. y_i takes +1 or −1 ($y_i \in \{+1, -1\}$, in this experiment, and y_i takes 1 or 0), indicating that the vector belongs to this class or not. It is said to be linearly separable if there is a linear function that can completely separate the two classes; otherwise it is nonlinearly separable.

Figure 3 is a linear separable case, since a straight line can be drawn to separate the vector of class +1 from the vector of class −1. There are countless such lines, and the so-called optimal classification line requires that the two samples be correctly separated and that the separation interval be the largest. SVM completes the classification of the sample by searching for the one that has the largest classification interval. The optimal classification line can be expressed by the equation $\omega \cdot x + b = 0$ ($\omega \in R^n$, $b \in R$); ω is the weight vector and b is the scalar, called the bias. The points above the separation hyperplane are satisfied

$$\omega \cdot x + b > 0. \tag{8}$$

Similarly, the points below the separation hyperplane are satisfied

$$\omega \cdot x + b < 0; \tag{9}$$

we can adjust the weight to make the edge side of the hyperplane able to be expressed as

$$H1 : \omega \cdot x + b \geq 1, \quad \text{for } y_i = 1$$

$$H2 : \omega \cdot x + b \leq 1, \quad \text{for } y_i = -1. \tag{10}$$

This means that the vectors falling on or above $H1$ belong to class +1 and the vectors falling on or below $H2$ belong to −1. From (10) we can get

$$y_i (\omega \cdot x + b) \geq 1, \quad \forall i. \tag{11}$$

Any of the training tuples falling on $H1$ and $H2$ are support vectors, and the equal sign is established.

From the above, we can get that the maximum edge is $2/\|\omega\|$. Finding that the maximum value of $2/\|\omega\|$ is equivalent to calculating the minimum value of $\|\omega\|$. Generalized to n-dimensional space, how the SVM finds the optimal hyperplane is equivalent to solving the constrained optimization problem; the formula is expressed as

$$\begin{aligned} &\min_{w,b} \quad \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^{N} \xi_i \\ &\text{s.t.} \quad y_i (\omega \cdot x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \ i = 1, \ldots, N, \end{aligned} \tag{12}$$

where $C > 0$ is the penalty parameter, indicating the degree of attention to the outliers, and the relaxation variable ξ_i is a measure of the degree of outliers [16].

DDoS attack detection is equivalent to two-classification problem; we use the SVM algorithm characteristics, collect switch data to extract the characteristic values to train, find the optimal classification hyperplane between the normal data and DDoS attack data, and then use the test data to test our model and get the classification results.

3. Experiment and Analysis

In this experiment, the controller (Floodlight [17]) and the switch (Openflow switch) are deployed under Ubuntu to generate the network topology diagram in Figure 4. The experimental topology is generated by mininet. The validity of DDoS attack detection method is verified by deploying SDN environment. PC1 and PC2 are the bot hosts; PC5 is the victim target. PC1 and PC2 can send normal packets to generate normal samples or send DDoS attack packets to generate DDoS attack samples. PC3 and PC4 generate normal network traffic samples. These samples are used for training to generate model and detecting attack.

During the training sample phase, the normal traffic is generated by PC3 and PC4. It includes TCP traffic, UDP traffic, and ICMP traffic. We use the classic DDoS attack tool Hping3 to generate abnormal network traffic. Hping3 is fully scriptable using the TCL language and can receive and send data packets by describing the binary or string representation of the data packets. In practice this means that a few lines of code can perform things that usually take many lines of C code. Examples are automated security tests with pretty printed report generation, TCP/IP test suites, many kind of attacks, NAT-ting, prototypes of firewalls, implementation of routing protocols, and so on. The advantage of hping3 is the ability to customize parts of the packet, so users can

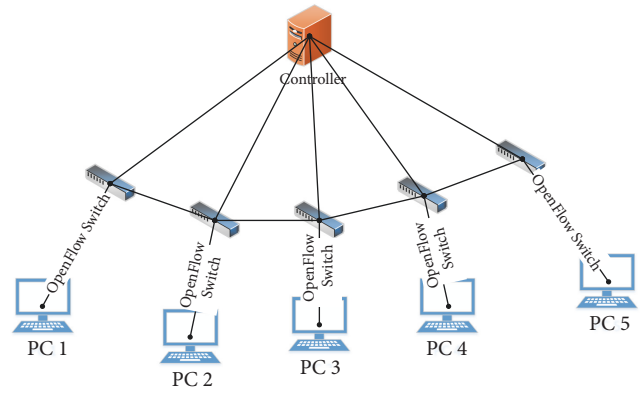


Figure 4: Network topology.

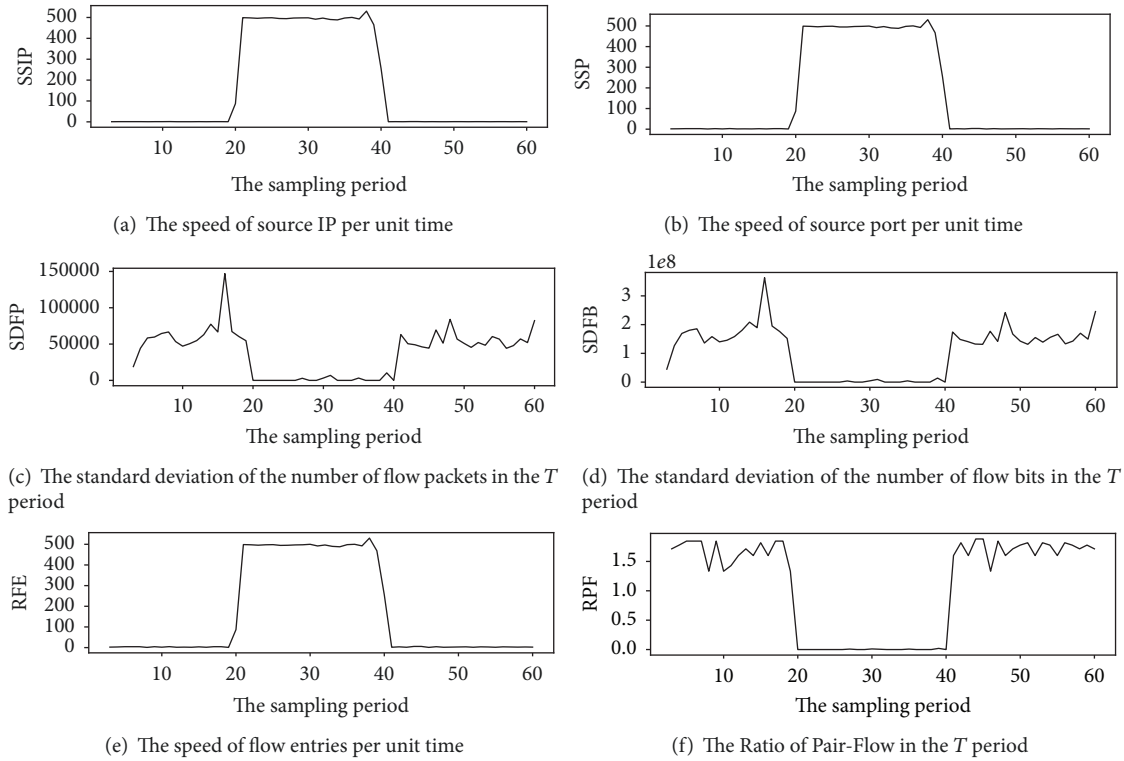
Table 1: The training and detection of attack flow samples.

Attack types	Training	Detection
TCP(200) flood		>30000
TCP(600) flood		>30000
TCP(1000) flood		>30000
UDP(200) flood		>30000
UDP(600) flood	>30000	>30000
UDP(1000) flood		>30000
ICMP(200) flood		>30000
ICMP(600) flood		>30000
ICMP(1000) flood		>30000

flexibly attack and detect the target [18]. Based on the above characteristics, we use Hping3 to generate different types of attack data. We use it to simulate the typical network traffic attack TCP SYN flood, UDP flood, and ICMP flood. These floods are used as training and for detection of attack samples. The types of attacks and the number of flows are shown in Table 1. The numbers in brackets are the size of the packets at the time of attack. They are same as the size of the packets of training data. We use the training data to generate the model. The training model is used to detect different attack data.

In this experiment, the sampling period T (interval) is 3 s. We attack in the $T20$ to $T40$ periods. During the sampling process, we collect the flow table data of 60 periods in the Openflow switch, then process and normalize the data of each period, and get the normal samples and DDoS attack flow samples of the six-tuple characteristic values matrix. The trends of the six-tuple characteristic values in 60 periods are shown in Figure 5.

In Figure 5, the abscissa represents period and the ordinate indicates the speed of source IP in a unit time (Figure 5(a)), the speed of source port in a unit time (Figure 5(b)), the standard deviation of the number of flow packets in the T period (Figure 5(c)), the standard deviation of the number of flow bits in the T period (Figure 5(d)), the speed of flow entries in a unit time (Figure 5(e)), and the Ratio of Pair-Flow in a T period (Figure 5(f)). In the experiment, we attack the $T20$–$T40$ periods. In the event of an attack, the number of flow entries per unit time will increase dramatically.



(a) The speed of source IP per unit time

(b) The speed of source port per unit time

(c) The standard deviation of the number of flow packets in the T period

(d) The standard deviation of the number of flow bits in the T period

(e) The speed of flow entries per unit time

(f) The Ratio of Pair-Flow in the T period

Figure 5: Six-tuple eigenvalue trend.

Generally, the attack is based on the pseudosource random IP addresses and port numbers. The amount of source IP and the number of source ports are also increased in a unit time. So there are similar growth trends in Figures 5(a), 5(b), and 5(e). Under normal circumstances, sending the data packets is relatively large, and in the attack, in order to achieve the attack effect, attacker usually sends data as soon as possible, so the data packets are relatively small and unchanged. Thus, the standard deviations of the number of flow packets and the number of flow bits in a T period are relatively small and have tiny fluctuations. As shown in Figures 5(c) and 5(d), the two characteristic parameters are large and fluctuating obviously in the normal periods, and they are very small and change gently in the $T20$–$T40$ periods. When we access the network normally, the source host and the destination host will produce interactive flow entries. In the time of an attack, due to using virtual random source IP addresses and source port numbers commonly, when the large amount of requests occur, the destination host cannot respond timely. Therefore, the proportion of interactive flow will decrease sharply. As shown in Figure 5(f), in the $T20$–$T40$ periods, the interactive flow entries drop to almost zero. Under the normal circumstances, the ratio of interactive flow entries is relatively large and fluctuates in a normal range.

We used the SVM function in Rstudio [19] to train the data to get the SVM model and use the model to predict the test data. We use the two characteristic values SSIP and RPF in the test data to draw classification chart; the classification results are shown in Figure 6.

In the experiment, the experimental data is nonlinear separable, and it is multidimensional, so the classification hyperplane is not a straight line or a plane but a curved surface (two-dimensional image displays curve). The light green area is the normal network access data. The pink area indicates that the network is being attacked. The red marks are the data distribution of the network being attacked. "×" represents the support vectors in this figure.

The performance of the attack detection is displayed by the detection rate (DR) and false alarm rate (FAR); the formulas are calculated as the values:

$$\mathrm{DR} = \frac{\mathrm{DD}}{\mathrm{DD} + \mathrm{DN}}. \tag{13}$$

In this formula, DD indicates that the attack flow is detected as an attack flow, and DN means that the attack flow is detected as a normal flow.

$$\mathrm{FAR} = \frac{\mathrm{FD}}{\mathrm{FD} + \mathrm{TN}}. \tag{14}$$

In the formula, FD means that the normal flow is detected as an attack flow, and TN indicates that the normal flow is detected as a normal flow.

In the experiment, the normal traffic is composed of three basic communication kinds of traffic (TCP, UDP, and ICMP) and the attack traffic consists of three separate types of attack traffic: TCP, UDP, and ICMP. The accuracy rate and false alarm rate of packet detection for different lengths of the three types of attack traffic are shown in Table 2. The average detection accuracy rate of this experiment is 95.24%, and the average false alarm rate is 1.26%, and the expected effect was achieved. The low false alarm rate is a good result and, on the other hand, it may be that our simulation of normal

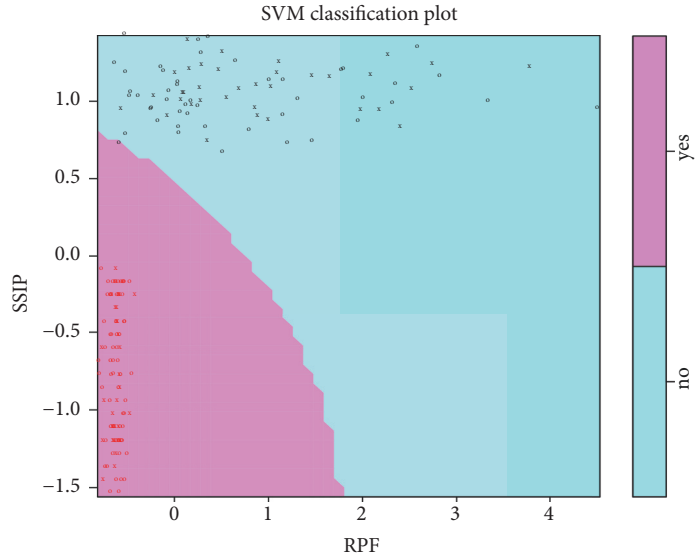


Figure 6: Classification results.

Table 2: The experimental results of three kinds of attacks.

	TCP			UDP			ICMP		
Packets size	200	600	1000	200	600	1000	200	600	1000
Detection accuracy rate	95.24%	100%	95.24%	95.24%	95.24%	95.24%	90.48%	95.24%	95.24%
Average		96.83%			95.24%			93.65%	
Average detection accuracy rate					95.24%				
False alarm rate	0.0%	0.0%	0.0%	2.7%	0.0%	0.0%	5.88%	0.0%	2.77%
Average		0.0%			0.9%			2.88%	
Average false alarm rate					1.26%				

data flow is not comprehensive enough, which is what we need to improve in the future. The relatively low accuracy rate of ICMP flow detection may be due to the fact that the ICMP traffic has no source port and destination port, so the characteristic matrix is only 4 dimensions. But our experimental results still have a high detection accuracy rate, which reached our goal.

4. Concluding Remarks

In this paper, the flow status information of the network traffic is collected on the switch by the controller. We extracted the six-tuple characteristic values related to DDoS attack and then use the support vector machine algorithm to judge the traffic and carry out DDoS attack detection. We focus on the analysis of the changes of the characteristic values of traffic and verify the feasibility of this method by deploying the SDN experimental environment. The detection accuracy rate of the experiment is high and the false alarm rate is low, which has obtained our expected results. In comparison, the test detection accuracy rate of ICMP attack flow is relatively low. By analyzing the ICMP traffic, we have come to the conclusion that the ICMP flow has no source port and destination port, so SSP and RPF are zero, which makes the six-tuple characteristic values matrix change into four-tuple characteristic values matrix, whether attacked or not. But this has little effect on the experimental results, and our experiment has achieved the goal. On the other hand, due to the very low false alarm rate, we should simulate the normal data flow more comprehensively, which is what we need to improve in the future.

Conflicts of Interest

There are no conflicts of interest in this paper.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (nos. 61762030, 61462007).

References

[1] H. Zhang, Z. Cai, Q. Liu, Q. Xiao, Y. Li, and C. F. Cheang, "A surveyon security-aware network measurement in SDN," *Security and Communication Networks*, Article ID 2459154, 2018.

[2] J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, and J. Zheng, "Disrupting SDN via the data plane: a low-rate flow table overow attack," in *Proceedings of the 13th EAI International Conference on Security and Privacy in Communication Networks*, Niagara Falls, Canada, October 2017.

[3] Z. Cai, Z. Wang, K. Zheng, and J. Cao, "A distributed TCAM coprocessor architecture for integrated longest prefix matching, policy filtering, and content filtering," *IEEE Transactions on Computers*, vol. 62, no. 3, pp. 417–427, 2013.

[4] Y. Li, Z. Cai, and H. Xu, "LLMP: exploiting LLDP for latency measurement in software-defined data center networks," *Journal of Computer Science and Technology*, vol. 33, no. 2, pp. 277–285, 2018.

[5] H. Lin and P. Wang, "Implementation of an SDN-based security defense mechanism against DDoS attacks," in *Proceedings of the 2016 Joint International Conference on Economics and Management Engineering (ICEME 2016) and International Conference on Economics and Business Management (EBM 2016)*, Pennsylvania, Penn, USA, 2016.

[6] J. G. Yang, X. T. Wang, and L. Q. Liu, "Based on traffic and IP entropy characteristics of DDoS attack detection method," *Application Research of Computers*, vol. 33, no. 4, pp. 1145–1149, 2016.

[7] A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, pp. 385–393, 2016.

[8] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10)*, pp. 408–415, Denver, Colo, USA, October 2010.

[9] N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," *Arabian Journal for Science and Engineering*, vol. 42, no. 2, pp. 425–441, 2017.

[10] X. Wang, M. Chen, C. Xing, and T. Zhang, "Defending DDoS attacks in software-defined networking based on legitimate source and destination IP address database," *IEICE Transaction on Information and Systems*, vol. E99D, no. 4, pp. 850–859, 2016.

[11] J. Xia, Z. Cai, G. Hu, and M. Xu, "An active defense solution for ARP Spoo ng in OpenFlow network," *Chinese Journal of Electronics*, vol. 3, 2018.

[12] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," in *Proceedings of the 2015 International Conference on Computing, Networking and Communications, ICNC 2015*, pp. 77–81, Garden Grove, Calif, USA, February 2015.

[13] M. Alazab, "Profiling and classifying the behavior of malicious codes," *The Journal of Systems and Software*, vol. 100, pp. 91–102, 2015.

[14] H. F. Li, X. L. Huang, and Z. Q. Zheng, "DDoS attack detection method based on software definition network and its application," *Computer Engineering*, vol. 42, no. 2, pp. 118–123, 2016.

[15] X. Nguyen, L. Huang, and A. D. Joseph, *Machine Learning and Knowledge Discovery in Databases*, Springer, Berlin, Germany, 2008.

[16] W. U. Shao-Hua, S. B. Cheng, and H. U. Yong, "Web attack detection method based on support vector machines," *Computer Science*, 2015.

[17] L. I. Ning, Z. A. Hao, and L. I. Yan, "Implementation and simulation research on openflow network architecture [J]," *Computer & Network*, 2014.

[18] Hping3, http://www.hping.org/hping3.html.

[19] RStudio, https://www.rstudio.com.