PAPER

Joint Selfattention-SVM DDoS Attack Detection and Defense Mechanism Based on Self-Attention Mechanism and SVM Classification for SDN Networks

Wanying MAN[†a)], **Guiqin YANG**[†b)], ***and*** **Shurui FENG**[†c)], ***Nonmembers***

SUMMARY Software Defined Networking (SDN), a new network architecture, allows for centralized network management by separating the control plane from the forwarding plane. Because forwarding and control is separated, distributed denial of service (DDoS) assaults provide a greater threat to SDN networks. To address the problem, this paper uses a joint high-precision attack detection combining self-attentive mechanism and support vector machine: a trigger mechanism deployed at both control and data layers is proposed to trigger the initial detection of DDoS attacks; the data in the network under attack is screened in detail using a combination of self-attentive mechanism and support vector machine; the control plane is proposed to initiate attack defense using the OpenFlow protocol features to issue flow tables for accurate classification results. The experimental results show that the trigger mechanism can react to the attack in time with less than 20% load, and the accurate detection mechanism is better than the existing inspection and testing methods, with a precision rate of 98.95% and a false alarm rate of only 1.04%. At the same time, the defense strategy can achieve timely recovery of network characteristics.

key words: *SDN, DDoS, self-attention, SVM, selfattention-SVM*

1. Introduction

The Software Defined Networking (SDN) architecture consists of three planes and two interfaces [1]. The southbound and northbound interfaces, which connect the three planes of the application plane, data plane, and control plane, respectively. Due to its flexibility, automation and openness, SDN networks are also becoming more widely used [2]. The most common and straightforward network attacks technique that continually poses a danger to the security of the information world is the Distributed Denial of Service (DDoS) attack. [3]. The fundamental idea is that the attacker infiltrates the system and network by flooding the server with a large number of malicious requests from a large number of terminals that have been taken over. The server is unable to offer common users the services they anticipate as a result [4].

However, the characteristics of SDN network architecture also lead to more security risks for SDN networks compared to traditional network architectures [5]. Therefore, DDoS attacks under SDN networks can be divided into two categories:
(1) Attacks against terminals;
(2) Attack against the central controller.

A single point of failure results in the loss of the network's fundamental functionality when the controller is attacked and cut off from the other network nodes [6]. The huge amounts of attack traffic are hidden within legitimate traffic and disguised as normal network data floods, so the entire network loses basic service capability [7]. The purpose of the attack against the endpoints in the SDN network is still to drain their resources so that they cannot provide normal services to legitimate users.

How to improve the SDN network to deal with DDoS attacks has become the key and the focus of SDN network development [8]. In the traditional network architecture, the detection and defense for the DDoS attacks are relatively mature. This causes the central controller to consume additional computational load and the southbound interface channel to add significant communication overhead. As a result, this work proposes a detection model. When compared to earlier detection approaches, the algrithm in this model fully utilizes the computing resources, boosts detection efficiency, and enhances detection accuracy. The following is a summary of this paper's contribution:
(1) A threshold-based triggering mechanism deployed at network nodes and controllers, respectively, is proposed for two attack types of DDoS attacks under SDN networks.
(2) We proposed Selfattention-SVM method innovatively combines the self-attention mechanism and SVM classification algorithm applied to attack detection in SDN networks.
(3) A cooperative defense mechanism is designed to implement screening and defense for attack traffic and normal network burst traffic at the forwarding and control layers to protect the network resources as well as the computational resources of the controller.
(4) Extensive experiments are run on a simulated network to assess the efficiency and efficacy of the suggested approach. The outcomes demonstrate that the proposed technique works well in terms of resource use and detection precision.

2. Related Work

The most used method of attack detection in SDN networks is still based on threshold [9]. An SDN-based technique

Manuscript received May 19, 2023.
Manuscript revised August 4, 2023.
Manuscript publicized September 5, 2023.

[†]The authors are with School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou, 730070 China.
a) E-mail: man435881972@outlook.com
b) E-mail: guiqiny2023@163.com (Corresponding author)
c) E-mail: 18893828750@163.com

DOI: 10.1587/transfun.2023EAP1057

called FlexProtect was presented by Chen et al. [10] to detect DDoS assaults by assessing the rate of open TCP connections. Meanwhile, Wang et al. [11] presented a threshold-based mechanism called Woodpecker, which assigns a score to each link in order to identify link flooding attacks. In the study by Gkountis et al. [12], a lightweight algorithm was designed to defend against DDoS attacks in SDNs.

Machine learning can be applied to network control thanks to SDN's novel application layer [13]. A DDoS detection and defense mechanism were put forth by Cui et al. [14]. It calculates the information entropy of the captured data to train the SVM defense model. A brand-new SOM-based DDoS detection method called DSOM was proposed by Phan et al. [15]. A single SOM performs the same functions as a DSOM module. It trains by extracting and transmitting part of the data. Wang et al. [16] presented a security guard strategy (SGS) to safeguard the SDN control plane against DDoS attacks with BPNN trained by the extracted attributes.

Machine learning-based attack detection mechanisms can achieve good results, but also increase power consumption and load. The long response time also leads to the network not reacting to the attack in time [17]. As a result, the joint detection is gaining more favor and some progress is being made.

Threshold-based DDoS detection can be used as a main rough detection approach or as a trigger mechanism. A detection technique based on a threshold and SVM combination was described by Yang et al. [18]. Prior to using the SVM-based DDoS attack detection approach, the IP entropy of the packets is first established.

Another idea is the combination of multiple machine learning methods. Initial screening or status determination using simple algorithms with lower accuracy before detection using higher accuracy detection mechanisms [19]. Phan et al. [20] developed an SDN-based defensive system for DDoS assaults in the cloud. For measuring attack checking, it combines SVM and self-organizing mapping (SOM). K-Means and KNN were used by Tan et al. to construct a DDoS detection system [21]. It consists of a KNN-based traffic detection module and a K-Means-based training data processing module.

3. Detection and Defense of Attacks on SDN Networks

We suggest a trigger mechanism that works together at the data plane and control plane to address the problem mentioned above. It uses separate triggers for endpoint and server attacks, while the combined algorithm of Self-Attention mechanism and Support Vector Machine deployed in the controller will precisely detect suspicious traffic after the trigger. If they are attacked, the defense mechanism of the attack is activated. Figure 1 depicts the model's overall structure. The main body of the control plane is the SDN controller, and the data plane contain several switches and hosts. The triggering mechanism of the two planes starts extracting traffic characteristics after detecting an attack, and classifies them through a precise detection mechanism. The controller then issues a defense strategy against the attack. The model is discussed in detail in this section.

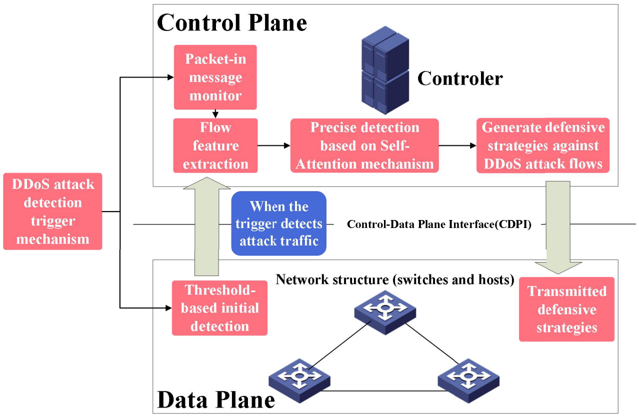

Control Plane
Packet-in message monitor
Controler
Flow feature extraction
Precise detection based on Self-Attention mechanism
Generate defensive strategies against DDoS attack flows
DDoS attack detection trigger mechanism
When the trigger detects attack traffic
Control-Data Plane Interface(CDPI)
Network structure (switches and hosts)
Threshold-based initial detection
Transmitted defensive strategies
Data Plane

Fig. 1 Selfattention-SVM detection defense mechanism model structure.

3.1 DDoS Attacks Detection Triggering Mechanism in the Data Plane and Control Plane

Some researchers have opted to integrate the controller's trigger mechanism. However, this would exhaust the controller's computational resources as well as the southbound interface's communication capabilities. Furthermore, the attack traffic will arrive at the switch first [22]. So we deployed triggering mechanisms on the switches and the controller.

The data plane trigger mechanism enables the initial recognition of passing communications. The information entropy-based trigger method is mostly used to identify terminal threats. The quantity of all matched packets is E_{in} reflected in the flow table entry when a packet matches an entry. We use Eq. (1) to obtain the number of packets in unit time Δt.

$$V_{in} = E_{in}(t + \Delta t) - E_{in}(t) \tag{1}$$

Where V_{in} is the amount of variation in matched packets within Δt, E_{in} is the number of packets matching the flow table entries in the flow table as of moment t.

We use the destination address as the main indicator of information entropy. Then the set of the number of destination IP addresses N is obtained, and

$$N = \{N_1, N_2, N_3, \ldots, N_i, \ldots, N_n\} \tag{2}$$

The total number of IP addresses of the ith endpoint in all packets matching s flow table entries in time Δt is Eq. (3). Then the probability of appearance of IP address of destination terminal i is shown in Eq. (4).

$$N_i = \sum_{h=1}^{s} V_{in_h} \tag{3}$$

$$P_{in} = \frac{N_i}{\sum_{i=1}^{n} N_i} \tag{4}$$

The information entropy is then determined using the Shannon entropy formula. The Shannon entropy is decreased during a DDoS assault because an exceptionally high number of packets share the same geographical address. Traffic is regarded as attack data when the entropy value is below the threshold.

When attacks are launched against the controller, we compute the packet-in data using monitoring to accomplish detection. There is a lot of traffic with fake IPs in the attack flow. The switch is unable to determine the traffic's forwarding path since they invariably do not correspond to the entries in the flow table [23]. The switch will now gather the traffic statistics and deliver it as a packet-in message to the controller. When the controller receives the packet-in message, it publishes the flow table. The packet-in message allows the controller to recognize attacks on its own system.

3.2 Relevant Features of the Precise Detection Method

Since the purpose of attack traffic is to attack, resulting in its own characteristics are very different from normal traffic [24]. This section will be based on an examination of the DDoS attack characteristics and algorithm characteristics, and it will choose the traffic features that aid to increase identification accuracy.

DDoS attack flows are short-lived and heavily asymmetric, so the proportion of asymmetric traffic in the network can grow rapidly. Therefore, asymmetric traffic in the network becomes an important indicator of DDoS attacks. The attack uses a different number of packets than regular traffic [25]. In order to quickly consume the bandwidth of the network link, attackers encapsulate the traffic with too many packets.

Attackers generally use the same IP protocol to attack the network, which will reduce the randomness of the network IP protocol [26]. Therefore, the entropy value of IP protocols is also an important indicator. The calculation results of the trigger mechanism deployed in the forwarding layer switches discussed earlier are also a distinctive feature, leveraging them as a data source for accurate detection algorithms.

After the above analysis, we use the following features as the base data for the detection algorithm in the controller:
(1) Average number of packets per flow (anpf): Given that the amount of attack traffic packets differs significantly from that of regular traffic, this is a crucial indicator of attacks. In Eq. (5) $N_{flow_{\Delta t}}$ is the total number of flows in Δt and the N_p is the number of packets in every flow.

$$anpf = \frac{\sum_{i=1}^{N_{flow_{\Delta t}}} N_p}{N_{flow_{\Delta t}}} \tag{5}$$

(2) Average flow hold time (afht): When a DDoS attack takes place, the duration of each stream will either be very long or very short. In Eq. T_{flow} denotes the holding time of single flow.

$$afht = \frac{\sum_{i=1}^{N_{flow_{\Delta t}}} T_{flow}}{N_{flow_{\Delta t}}} \tag{6}$$

(3) Average bytes per second (abps): Due to the attack flow's high or low byte density compared to regular traffic, the amount of bytes passing through the switch will differ from normal traffic. t_n is the current moment, t_{n-1} indicates the previous moment. Meanwhile, b_{t_n} is the number of bits at the current moment and $b_{t_{n-1}}$ means the number of bits at the previous moment.

$$abps = \frac{b_{t_n} - b_{t_{n-1}}}{t_n - t_{n-1}} \tag{7}$$

(4) Asymmetric flow ratio per second (afps): Due to the significant increase in attack asymmetric traffic, the percentage of asymmetric traffic during the attack will increase dramatically. In Eq. (8), $N_{sflow_{\Delta t}}$ and $N_{aflow_{\Delta t}}$ represent the number of streams with symmetric and asymmetric flow in Δt.

$$afps = \frac{\sum_{i=1}^{N_{flow_{\Delta t}}} N_{aflow_{\Delta t}}}{\sum_{i=1}^{N_{flow_{\Delta t}}} N_{aflow_{\Delta t}} + \sum_{i=1}^{N_{flow_{\Delta t}}} N_{sflow_{\Delta t}}} \tag{8}$$

(5) The entropy of the protocol of the destination IP per second (epipps): A large amount of attack traffic uses the same protocol on the same target IP, so the protocol entropy of the target IP should be significantly lower than that of normal traffic [27]. Similar to the way the target IP entropy is calculated for the trigger mechanism.
(6) The entropy of the destination IP per second (edipps): This comes from the control surface trigger mechanism, through which it remains an important indicator, and we will use this data again.

For precise detection, the controller computes and stores the information properties of the pertinent flow rate using the aforementioned algorithm.

3.3 DDoS Accurate Detection Algorithm

The primary control functions of the controller and the resources of the detection algorithm must be balanced when the controller deploys an accurate detection mechanism for attacks with DDoS [28]. Although the complexity of the previous method ensures guaranteed computational accuracy, it requires longer computational time and larger computational resources, which leads to a decline in basic network control functions [29]. For this purpose, we employ the Selfattention-SVM mechanism, which is a combination of Self-Attention mechanism and SVM, to detect DDoS attacks.

Our proposed accurate detection method mainly consists of a Self-Attention mechanism traffic feature correlation extraction module and a SVM traffic binary classification module. After the controller extracts the suspicious traffic features of the switch, multiple features are combined into a sequence of feature vectors x, which is used as the main computational data of the Self-Attention mechanism. Each vector of the input feature sequence data, calculates its own similarity to all other vectors separately to obtain a new vector. Due to the use of higher dimensional sequences for training as well as computation, too high a feature dimension can lead to too large a similarity value being computed, which further leads to the saturation of the normalization function SoftMax. We choose to use Eq. (8) to calculate the similarity of the feature sequences.

$$S = \frac{\langle Q, K \rangle}{\sqrt{D_x}} \tag{9}$$

where x is the input feature sequence, D_x the dimension of the input feature sequence, Q and K are the interrogation sequence and key sequence in the attention mechanism, respectively. In the attention mechanism, both Q and K are features that need to be learned and obtained. Since we use the self-attention mechanism, here both the query sequence and the key sequence are the input feature sequence X, so the similarity calculation formula is deformed to

$$S_{feature} = \frac{\langle X, X \rangle}{\sqrt{D_X}} \tag{10}$$

After that, the calculated similarity vectors are normalized by the SoftMax function to obtain the similarity weights between the vectors. Finally, the feature mapping is obtained by calculating the similarity between the input sequence and the weight vectors again. The feature mapping has been sufficiently extracted to the correlation between each feature vector. Due to the input vector independence and time independence of the self-focus mechanism, the self-focus mechanism has the ability of parallel computation, which improves the computational speed. Thus, the equation of the Self-Attention mechanism can be expressed as:

$$H = Selfattention = Softmax\left(\frac{\langle X, X \rangle}{\sqrt{D_X}}\right) \tag{11}$$

If both the input and the output of the Self-Attention model are equal-length sequences of length m, so H is a sequence of vectors of length m.

The feature mapping at this point will be classified using the SVM algorithm. We choose to use a linear kernel function $K(x, y) = x^T y + 1$ to map each vector in the sequence of feature mappings output by the Self-Attention mechanism to a higher dimensional space, at which point the optimization objective of the classification algorithm is:

$$min\left(\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K\left(x_i, x_j\right) + C\sum_{i=1}^{N}\varepsilon_i\right) \tag{12}$$

where C is a hyperparameter indicating how much importance we place on "correct classification", ε_i is the relaxation factor and satisfies $\varepsilon_i \geq 0$. After solving using the Lagrange multiplier method, formulas (13) and (14) are used as classification functions when classifying:

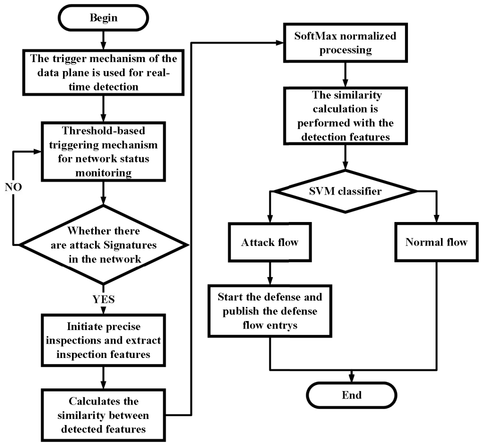

Begin
The trigger mechanism of the data plane is used for real-time detection
Threshold-based triggering mechanism for network status monitoring
NO
Whether there are attack Signatures in the network
YES
Initiate precise inspections and extract inspection features
Calculates the similarity between detected features
SoftMax normalized processing
The similarity calculation is performed with the detection features
SVM classifier
Attack flow
Normal flow
Start the defense and publish the defense flow entrys
End

Fig. 2 Selfattention-SVM calculation flow chart.

$$y = \begin{cases} 1, & d > 0 \\ -1, & d < 0 \end{cases} \tag{13}$$

$$d = \sum_{i=1}^{N}\left[\alpha_i y_i K\left(x, x_i\right)\right] + b_0 \tag{14}$$

where b_0 is the parameter obtained by the following equation:

$$b_0 = y_i - \varepsilon_i y_j K\left(x_i, x_j\right) \tag{15}$$

Finally, we use the classification function to classify the traffic. An output of 1 indicates that the traffic is normal, and −1 represents DDoS attack traffic. Figure 2 shows the basic computational flow of Selfattention-SVM.

3.4 Attack Defense Mechanism

When attacks are detected, appropriate countermeasures must be taken as soon as possible [30].

By sending a "Flow-mod" message to the switch when the defense is launched, the controller modifies the flow table entries in the switch. The "Command" field in the "Flow-mod" message is set to "ADD" by the controller. As a result, the priority of the entry in the current flow table that corresponds to the source IP information on the incoming port of the attacked traffic is increased and a new flow table entry is set to be dropped. To match the item in the defense flow table, the DDoS attack traffic can be inserted with a priority. At the same time, the controller sets the command field to "DELETE", causing the switch to delete malicious flow table entries that already exist due to DDoS attacks, freeing up occupied resources in the switch. By setting the "idle_time" field in the "Flow-mod" message to delete the defensive flow table entries in the switch once no attack traffic is matched for a predetermined amount of time, the controller also makes sure that the switch ceases the defense.

If the detection mechanism determines that the traffic is legitimate, the controller will not activate the defense

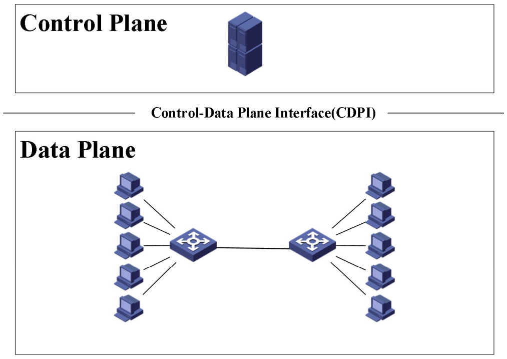


Fig. 3 Experimental topology.

mechanism.

4. Experimentation and Evaluation

4.1 Experimental Environment

The experiment uses RYU as the SDN controller and Mininet to simulate the network architecture. The control plane is a computer running the RYU controller. The CPU configuration of the experimental server is Intel (R) Core (TM) i7-8700K processor, the graphics card is Nvidia 1070ti, 16 GB RAM, and the operating system is Ubuntu18. The SDN network data plane was also configured in another computer using a virtual machine using Mininet, which included the relevant links, two SDN switches, and 10 hosts, each connected to five hosts respectively. The topology of this experimental environment is shown in Fig. 3.

In the experiments, a unit time of 1 s was selected and CICDDoS2019 [31] was used as the dataset for model training and validation. This dataset is a dataset containing both labeled and unlabeled network traffic and is mainly used to evaluate the effectiveness of DDoS detection algorithms and network security defense systems. In this experiment, four common DDoS attack types in the CICDDoS2019 dataset were used: UDP Flood, HTTP Flood, SYN Flood and ICMP Flood attacks. These types of attacks can not only represent the main threats in the current network environment, but also represent a significant challenge for dynamic security control of SDN networks. In addition, the data plane is attacked by the Kali system and Scapy is used to send large amounts of legitimate data to simulate legitimate traffic spikes to test the feasibility and effectiveness.

4.2 Training and Evaluation with Datasets

It is necessary to distinguish four attack flows (SYN Flood, ICMP Flood, UDP Flood and HTTP Flood) from the CICIDS2019 dataset, and block each attack flow independently, and use the filtering function to filter out records with specific flows. For each attack flow, a stratified sampling method is used for blocking so that the ratio of each attack stream is the same as the original data set. In each attack stream, the corresponding number of attack streams are randomly selected in the ratio of 70:15:15 and recorded into the training set, validation set and test set, respectively. Table 1 shows the number of stream records used in this experiment. In this paper, The Selfattention-SVM model is trained using the training set, validated using the validation set to avoid overfitting, and tested using the test set. Results of model tests are assessed using the table displayed below.

Table 1 Model experiment data.

Type of attack	Train set	Validation set	Test set
UDP Flood	2925	625	625
ICMP Flood	927	207	207
SYN Flood	20772	4439	4439
HTTP Flood	15034	3209	3209

(1) Precision:

$$precision = \frac{TP}{TP + FP} \tag{16}$$

is the proportion of the total number of samples predicted as positive samples to the amount of samples that the classifier correctly identified as positive samples. The more accurate the metric, the more attacks can be detected.

(2) Recall:

$$Recall = \frac{TP}{TP + FN} \tag{17}$$

denotes the ratio of correctly judged all attacked samples, i.e., the proportion of positive samples that are accurate to those that are overall positive. The higher the recall, the better the classifier is at identifying attacks.

(3) F1 score:

$$F1 = 2 \times \frac{Precision \times Recall}{Precision \times Recall} \tag{18}$$

it is a combined performance metric that takes into account accuracy and recall and is a weighted sum average. The performance improves as the F1 score rises.

(4) False Alarm Rate:

$$FPR = \frac{FP}{FP + TN} \tag{19}$$

demonstrates the proportion of non-attack samples that have been mistaken as attacks to all non-attack samples. A lower rate of false alarms suggests an improved understanding of normal flows.

Among the above evaluation metrics, TP (True Positive): indicates the amount of attack traffic that the model correctly classifies as attack traffic. FP (False Positive): indicates the number of normal traffic that the model incorrectly identifies attack traffic. TN (True Negative): shows how many normal flows the model correctly classifies as normal. FN (False Negative): demonstrates how many attack flows the model erroneously interprets as normal. The better the model performs, the higher the precision, recall, and F1

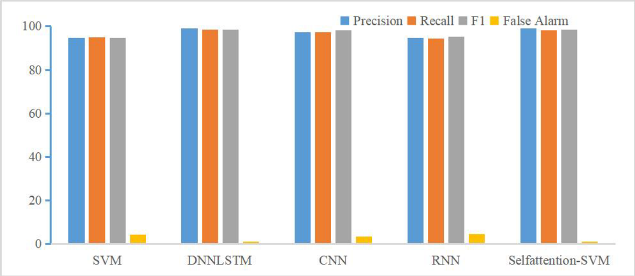


Fig. 4 Comparison among the model's precision, recall, F1, and false alarm rates with various algorithms.

Table 2 Comparison of the effects of various methods.

Methods	Precision	Recall	F1	False Alarm Rate	Sources
SVM	94.75%	94.84%	0.9456	4.24%	[14]
DNNLSTM	99.35%	98.64%	0.9856	0.63%	[33]
CNN	97.34%	97.47%	0.9815	2.35%	[32]
RNN	94.67%	94.47%	0.9525	4.68%	[34]
Selfattention-SVM	98.95%	98.08%	0.9846	1.04%	This paper

scores and the lower the false alarm rate.

The results of these methods are compared with the Selfattention-SVM technique developed in this research. In this paper, other DDoS attack detection algorithms are also employed for training tests using the same training set, validation set, and test set. Figure 4 shows that the DNNLSTM-based [31] method performs best and outperforms other methods, while the Selfattention-SVM method performs equally well in terms of precision, recall, and F1 score. A comparison of the various methods is shown in Table 2.

In the present research, we also pay particular attention to the model's effectiveness against four unique DDoS attacks. The model's performance in terms of precision, recall, F1 score, and false alarm rate when specific attack techniques are present is highlighted in Fig. 5. Because the training data of UDP Flood and ICMP Flood are less leading to the model's poor performance data in generalization ability for these two attacks, while SYN Flood and HTTP Flood are relatively more common attacks in the real network environment, so the training samples sampled from the dataset are large enough to ensure that the model learns more adequately for these two attacks, so the accuracy and recall data are better than the first two attacks, and the false alarm rate is relatively lower.

4.3 Performance Evaluation Using Simulated Traffic

In this experiment, we use the built SDN network topology to launch UDP Flood, ICMP Flood and SYN Flood attacks on the experimental topology from one endpoint in the virtual machine using Hping. As HTTP Flood assaults are launched using GoldenEye to haphazardly target different terminals, the effectiveness of Selfattention-SVM for actual traffic attacks is tested.

In the simulated attack, the proposed detection and defense mechanism uses a threshold-based trigger mechanism, so a composite detection mechanism based on threshold and SVM algorithm is used for comparison. The controller CUP occupancy rate is utilized as a key indicator of controller computing resources in the simulation because in SDN networks, ensuring that the controller has enough computing resources is essential for DDoS attack defense. The CUP usage of the approach in this work and the SOM-SVM method are contrasted [19]. In the first 18 seconds or so when the attack is not present, as shown in Fig. 6, the occupancy rate of the controller of Selfattention-SVM is lower than that of the SOM-SVM method. However, the occupancy rate of both techniques soon rises to a greater level of more than 95%, while the attack traffic is generated in the network about 20 seconds. At this time, the detection and defense mechanism in the controller is running and CPU resources are heavily occupied, and the occupancy rates of the two methods are not very different. When the detection is completed, the proposed method in this paper makes the CPU occupancy rate return to the normal level quickly. Because the trigger mechanism is deployed in the network nodes and the controller, the precise detection mechanism in the controller starts to run only when an attack alert occurs, and does not require the data plane to upload malicious traffic information at all times

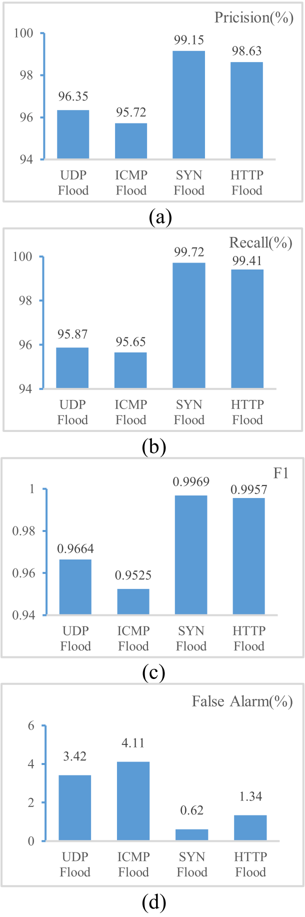


Fig. 5 The model compares the precision, recall, F1, and false alarm of the four attack methods.

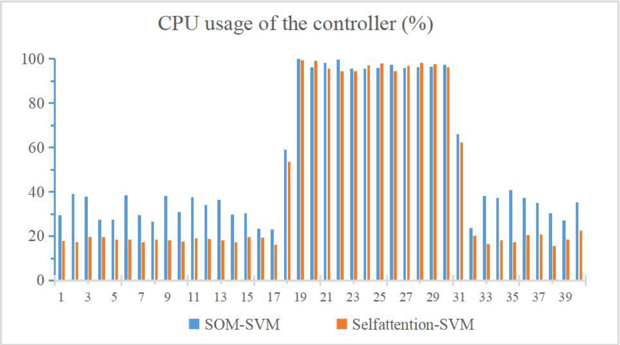
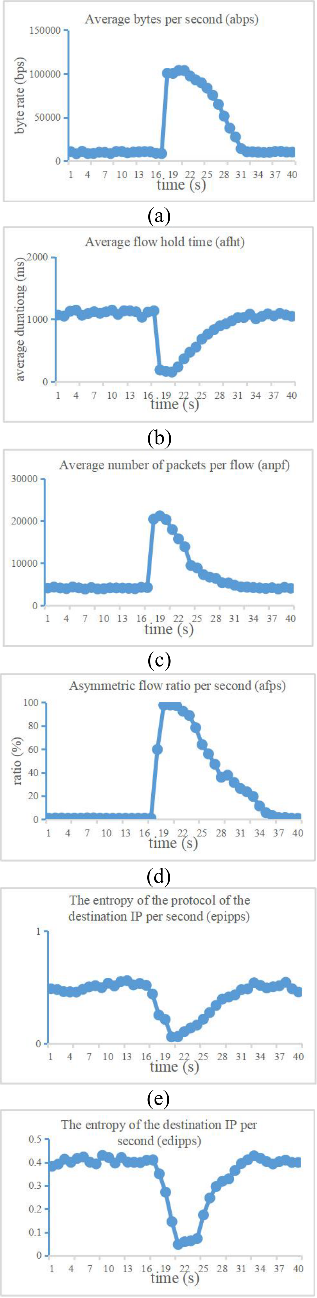


Fig. 6 CPU usage comparison.

for initial detection using the SVM method, so it does not take up too much of the controller's computational resources when no attack occurs, making the controller's computational resource usage better than the compared methods.

Based on the aforementioned attack simulation, the effectiveness of the defense of the approach proposed in this paper is evaluated by paying close attention to changes in the traffic features chosen during model detection. These traffic characteristics change accordingly when the attack does not occur, when the attack occurs, and after the attack is defended. Therefore, by observing the changes in the traffic characteristics, the defense strategy's efficacy against the attack traffic can be confirmed.

In the virtual network of the simulation experiment, a DDoS attack is launched to the network through host h1. As observed in Fig. 7, DDoS attack traffic begins to arrive in the network at roughly 16 s, which causes the metrics for all characteristics to alter as a result of the assault in a very short amount of time. At the same time, the controller receives a warning from the data plane triggering mechanism, which has detected the attack. The Selfattention-SVM precision detection mechanism is activated by the controller as soon as it gets the alert from the data plane in order to categorize and label the traffic. The controller also activates the defensive strategy in order to defend against the designated assault traffic. Therefore, after a few seconds of the attack traffic, the indicators of each characteristic gradually return to the normal level before being attacked. These six metrics demonstrate the success of the suggested defense strategy.



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 7 Changes in flow features.

5. Conclusion and Future Works

This paper fully analyzes the advantages and characteristics of using SDN network architecture, fully combines the traditional threshold detection method as the triggering mechanism, and combines machine learning methods to realize a security strategy that combines triggering, detection and defense against DDoS attacks. Experiments show that the trigger method proposed in this research may effectively and promptly detect attack traffic while ensuring the controller's computational resources. The defense system can successfully reduce the effects of DDoS attacks, and the accurate detection technique can successfully classify the attack traffic.

In this study, the self-attention mechanism is applied to SDN network attack detection. However, in the future, SDN network security must consider large-scale network models, and the combination of Self-Attention mechanism and other neural networks is more worth trying. The Transformer model, which deeply integrates the self-attention mechanism and the Feedforward Neural Network, will have more brighter future in this field. And more effective classification models should be explored in conjunction with it.

References

[1] "SDN Security Considerations in the Data Center," Open Networking Foundation, https://opennetworking.org/sdn-resources/solution-briefs/sdn-security-considerations-in-the-data-center/, accessed Aug. 11. 2022.

[2] G. Oluchi Anyanwu, C.I. Nwakanma, J.-M. Lee, and D.-S. Kim, "Optimization of RBF-SVM kernel using grid search algorithm for DDoS attack detection in SDN-based VANET," IEEE Internet Things J., vol.10, no.10, pp.8477–8490, 2023.

[3] A. Zainudin, L.A.C. Ahakonye, R. Akter, D.-S. Kim, and J.-M. Lee, "An efficient hybrid-DNN for DDoS detection and classification in software-defined IIoT networks," IEEE Internet Things J., vol.10, no.10, pp.8491–8504, 2023.

[4] T. Wang, H. Chen, and G. Cheng, "Research on software-defined network and the security defense technology," Journal of Communications, vol.38, no.11, pp.133–160, 2017.

[5] J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed, and S.A. Shah, "A time-efficient approach toward DDoS attack detection in IoT network using SDN," IEEE Internet Things J., vol.9, no.5, pp.3612–3630, 2022.

[6] A.S. Alqahtani, "FSO-LSTM IDS: Hybrid optimized and ensembled deep-learning network-based intrusion detection system for smart networks," J. Supercomput., vol.78, pp.9438–9455, 2022.

[7] P. Maity, S. Saxena, S. Srivastava, K.S. Sahoo, A.K. Pradhan, and N. Kumar, "An effective probabilistic technique for DDoS detection in OpenFlow controller," IEEE Syst. J., vol.16, no.1, pp.1345–1354, 2022.

[8] S. Tharewal, M.W. Ashfaque, S.S. Banu, P. Uma, S.M. Hassen, and M. Shabaz, "Intrusion detection system for industrial Internet of things based on deep reinforcement learning," Wireless Commun. Mobile Comput., vol.2022, pp.1–8, March 2022.

[9] L. Zhao, Z. Yin, K. Yu, X. Tang, L. Xu, Z. Guo, and P. Nehra, "A fuzzy logic-based intelligent multiattribute routing scheme for two-layered SDVNs," IEEE Trans. Netw. Service Manag., vol.19, no.4, pp.4189–4200, Dec. 2022, doi: 10.1109/TNSM.2022.3202741.

[10] M. Chen, J. Ciou, I. Chung, and C. Chou, "FlexProtect: A SDN-based DDoS attack protection architecture for multi-tenant data centers," Proc. International Conference on High Performance Computing in Asia-Pacific Region, pp.202–209, 2018.

[11] L. Wang, Q. Li, Y. Jiang, X. Jia, and J. Wu, "Woodpecker: Detecting and mitigating link-flooding attacks via SDN," Computer Networks, vol.147, pp.1–13, 2018.

[12] C. Gkountis, M. Taha, J. Lloret, and G. Kambourakis, "Lightweight algorithm for protecting SDN controller against DDoS attacks," 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC), Valencia, Spain, pp.1–6, 2017.

[13] M. Imran, M. Durad, F. Khan, and A. Derhab, "Toward an optimal solution against denial of service attacks in software defined networks," Future Generation Computer Systems, vol.92, pp.444–453, 2019.

[14] J. Cui, M. Wang, Y. Luo, and H. Zhong, "DDoS detection and defense mechanism based on cognitive-inspired computing in SDN," Future Generation Computer Systems, vol.97, pp.275–83, 2019.

[15] T. Phan, N. Bao, and M. Park, "Distributed-SOM: A novel performance bottleneck handler for large-sized software-defined networks under flooding attacks," Journal of Network and Computer Applications, vol.91, pp.14–25, 2017.

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

[17] N. Bawany, J. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: Methods, practices, and solutions," Arab. J. Sci. Eng., vol.42, no.2, pp.425–441, 2017.

[18] L. Yang and H. Zhao, "DDoS attack identification and defense using SDN based on machine learning method," 2018 15th International Symposium on Pervasive Systems, Algorithms and Networks (I-SPAN), Yichang, China, pp.174–178, 2018.

[19] Y. Cui, Q. Qian, C. Guo, G. Shen, Y. Tian, H. Xing, and L. Yan, "Towards DDoS detection mechanisms in software-defined networking," Journal of Network and Computer Applications, vol.190, p.103156, 2021.

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

[21] L. Tan, Y. Pan, J. Wu, J. Zhou, H. Jiang, and Y. Deng, "A new framework for DDoS attack detection and defense in SDN environment," IEEE Access, vol.8, pp.161908–161919, 2020.

[22] S. Kaur, K. Kumar, N. Aggarwal, and G. Singh, "A comprehensive survey of DDoS defense solutions in SDN: Taxonomy, research challenges, and future directions," Computers & Security, vol.110, p.102423, 2021.

[23] C. Chen, Z. Liao, Y. Ju, C. He, K. Yu, and S. Wan, "Hierarchical domain-based multicontroller deployment strategy in SDN-enabled space–air–ground integrated network," IEEE Trans. Aerosp. Electron. Syst., vol.58, no.6, pp.4864–4879, Dec. 2022, doi: 10.1109/TAES.2022.3199191.

[24] J. Singh and S. Behal, "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions," Computer Science Review, vol.37, p.100279, 2020.

[25] A. Akhunzada, E. Ahmed, A. Gani, M.K. Khan, M. Imran, and S. Guizani, "Securing software defined networks: Taxonomy, requirements, and open issues," IEEE Commun. Mag., vol.53, no.4, pp.36–44, April 2015.

[26] N. Aslam, S. Srivastava, and M. Gore, "ONOS flood defender: An intelligent approach to mitigate DDoS attack in SDN," Transactions on Emerging Telecommunications Technologies, vol.33, no.9, e4534, 2022.

[27] Y. Feng, R. Guo, D. Wang, and B. Zhang, "Research on the active DDoS filtering algorithm based on IP flow," 2009 Fifth International Conference on Natural Computation, 2009.

[28] A. Ahalawat, S.S. Dash, A. Panda, and K.S. Babu, "Entropy based DDoS detection and mitigation in OpenFlow enabled SDN," 2019 International Conference on Vision Towards Emerging Trends in Communication and Networking (ViTECoN), Vellore, India, pp.1–5, 2019.

[29] D. Advait, F. Hao, S. Mukherjee, T. Lakshman, and R. Kompella, "Towards an elastic distributed SDN controller," Proc. second ACM SIGCOMM workshop on Hot topics in software defined networking, pp.7–12, 2013.

[30] D. Hu, P. Hong, and Y. Chen, "FADM: DDoS flooding attack detection and mitigation system in software-defined networking," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, pp.1–7, 2017.

[31] I. Sharafaldin, A. Lashkari, S. Hakak, and A. Ghorbani, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," 2019 International Carnahan Conference on Security Technology (ICCST), 2019.

[32] J. Kim, J. Kim, H. Thi Thu, and H. Kim, "Long short term memory recurrent neural network classifier for intrusion detection," 2016 International Conference on Platform Technology and Service (PlatCon), pp.1–5, Feb. 2016.

[33] M.A. Razib, D. Javeed, M.T. Khan, R. Alkanhel, and M.S. A. Muthanna, "Cyber threats detection in smart environments using SDN-enabled DNN-LSTM hybrid framework," IEEE Access, vol.10, pp.53015–53026, 2022.

[34] C. Li, Y. Wu, X. Yuan, Z. Sun, W. Wang, X. Li, and L. Gong, "Detection and defense of DDoS attack-based on deep learning in OpenFlow-based SDN," International Journal of Communication Systems, vol.31, no.5, e3497, 2018.

Wanying Man received the B.S. in School of Electronics and Information, Xi'an Polytechnic University, China, 2019. And He is currently pursuing a M.Sc. degree in School of Electronic and Information Engineering, Lanzhou Jiaotong University, China. His research interest is Software Defined Network defense.

Guiqin Yang is currently a professor of School of Electronic and Information Engineering, Lanzhou Jiaotong University. She received her B.Sc. and M.Sc. degrees in communication and electronic system from Lanzhou Jiaotong University, China, in 1997. Her interested research fields are digital communication theory, network technology and signal processing.

Shurui Feng received a bachelor of engineering degree from Gansu University of Political Science and Law in 2018. She is currently pursuing a master's degree at Lanzhou Jiaotong University. Her current research direction is Software Defined Network defense.