Detection and Defense Against DDoS Attack on SDN Controller Based on Spatiotemporal Feature

Yan Xu[1,2](✉), Jinxing Ma[1], and Sheng Zhong[2]

[1] School of Computer Science and Technology, Anhui University, Hefei 230601, China
xuyan@ahu.edu.cn

[2] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China

Abstract. Software defined network (SDN) is an important part of the next generation computer network. The controller enables SDN to provide flexible data processing and programmable functions, which is the core of SDN. Once the controller is paralyzed, the whole network will be disrupted. DDoS attack targeting the controller will pose a great threat to SDN. However, most of the existing DDoS attack detection schemes only focus on the temporal or content feature of network data, it is easy to fail to detect attack or produce misjudgment. In this paper, we use the temporal and spatial feature of network data to detect DDoS attack on SDN Controller. Furthermore, flow table is used to defend against DDoS attack. We used the DARPA data set to perform experiments, and compared the performance with other scheme. The results show that our scheme can accurately detect DDoS attack and defend against it efficiently.

Keywords: SDN · DDoS attack · CNN-LSTM · Spatiotemporal feature · Network defense mechanisms

1 Introduction

Software-Defined Networking(SDN) is a new network architecture that can manage network traffic more effectively than traditional networks. By virtue of the separation of the control plane and forwarding plane, SDN can not only realize flexible network traffic regulation, but also easily accomplish advanced functions such as route management through programming. These functions can only achieve through complex device configurations in traditional network [1]. However, the introduction of phase separation technology between the control plane and forwarding plane also leads to some security issues. The centralization of logic function makes the SDN control plane vulnerable to malicious attacks, which in turn leads to a single point of failure [2]. Therefore, network security issues are seen as one of the most urgent problems in the SDN architecture [3].

S. Yu et al. (Eds.): SPDE 2020, CCIS 1268, pp. 3–18, 2020.
https://doi.org/10.1007/978-981-15-9129-7_1

DDoS attack on SDN Controller have received widespread attention since the controller is the core of SDN. In DDoS attack on SDN Controller, attackers constantly consume controller's resources in order to make the controller unable to provide normal services [4–6]. Eventually, if the controller cannot provide normal services, the entire network will be greatly affected or even paralyzed [7]. In recent years, many researchers have proposed some detection schemes to detect DDoS attack on SDN Controller. However, since the OF(OpenFlow) switch will send the packet to the controller after receiving the unknown packet, an attacker can use the OF switch to send the attack packet to the controller indirectly. This mechanism makes DDoS attack on SDN Controller difficult to detect.

This paper designed and implemented a scheme to detect and defense DDoS attack on SDN Controller. In this scheme, packets are first processed into samples that can reflect the change of traffic over a period of time [8,9]. For detection, the deep neural network built by Convolution Neural Network(CNN) and Long Short Term Memory(LSTM) will conduct attack detection on the generated samples. For defense, the scheme uses lightweight calculation to identify the attacker to block the attack traffic without affecting the normal service provided by the victim. It can be seen from the experimental results that the proposed scheme can accurately detect a variety of attacks type and block the attack traffic while making the victim continue to provide normal services. The contributions of this paper are summarized as follows:

- We design a novel preprocessing stage. At this stage, the scheme constructs samples that can reflect the spatial-temporal characteristics of the data flow by extracting features such as joint entropy and number of hosts that reflect the state changes of the data flow.
- We built a module containing a deep neural network model to detect DDoS attacks on the controller. As the samples can reflect the spatial and temporal characteristic of the data flow, the deep network model is constructed by CNN and LSTM. This is because CNN can effectively extract the spatial structure of data, and LSTM is the best choice when processing sequential data. As a result, the model can perform high-precision DDoS attack detection.
- A module to defend DDoS attack on SDN Controller is deployed in the scheme. The module uses lightweight computation to exactly determine the attacker's attributes based on the flow table information and install defensive flow entry to handle the attack packet so that the victim can still provide normal services.

The rest of this paper is organized as follows. Section 2 introduces the background knowledge of the scheme, and Sect. 3 mainly describes the related work. As for Sect. 4, it illustrates the details of the scheme. The evaluation result is provided in Sect. 5. Finally, in Sect. 6, this paper will be concluded.

2 Background

In this section, we will introduce the SDN architecture and OpenFlow protocol.

2.1 Software Defined Network

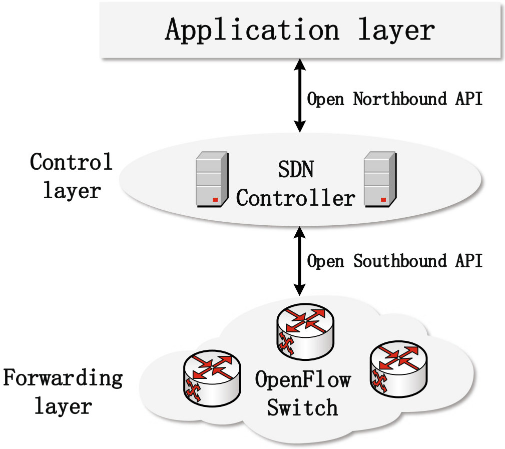

Fig. 1. SDN architecture

SDN has a layered structure as shown in Fig. 1. It consists of application, control and forwarding layer.

- Application layer: The application layer contains many applications that provide different functions. These applications communicate with controllers using northbound interface according to their network requirements [10].
- Control layer: The control layer is the brain of SDN, integrating all logic processing capabilities, which is the biggest difference between SDN and traditional network. This layer can program the network resources, update the forwarding rules dynamically and manage the network more flexibly than the traditional network. The main object of the control layer is the controller, which can generate network traffic operation instructions according to the requirements of various applications in the application layer, and sends the generated operation instructions to the forwarding layer through the southbound interface, indicating how the forwarding devices work.
- Forwarding layer: The forwarding layer consists of several OF switches. Different from the forwarding devices in the traditional network, OF switches can only forward the corresponding data packets according to the instructions sent by the controller, and has no logical processing function.

2.2 OpenFlow

OpenFlow is the protocol followed by the interaction between control layer and forwarding layer [11]. There is a flow table in each OF switch according to

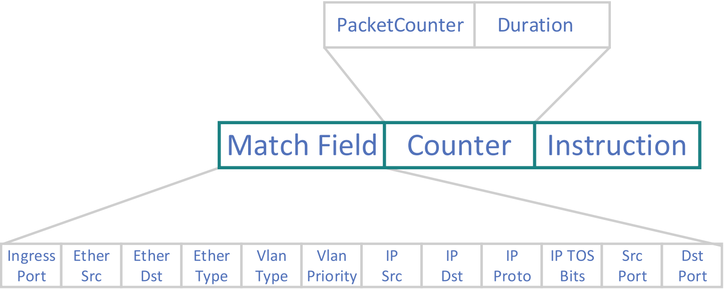

Fig. 2. Flow entry

OpenFlow 1.0. The flow table is composed of many flow entries, which instruct the data packets received by the OF switch to perform operations such as forwarding. Flow entry is mainly composed of *Match Fields*, *Counter* and *Instructions*. Each time a packet matches *Match Fields* content of a flow entry, the packet performs the actions contained in *instructions*. The function of the *Counter* is to count the number of packets that match the current flow entry and other statistics. The components of a flow entry are shown in Fig. 2.

Under the SDN architecture, when OF switch receives packet that do not match any flow entry within the flow table, it will send the packet to the controller. The controller first generate the processing action and then installs the action as a flow entry into OF switch to handle the mismatched packet. Due to this special mechanism, OF switch can perform fine-grained data flow processing compared to switch in traditional networks. However, attackers also can use the mechanism to launch DDoS attacks.

Since the controller integrates all the logical processing capabilities and can generate instructions to adjust the work of the forwarding device. Therefore, it is feasible to use the controller to obtain data flow and bring it into the established attack detection module for attack detection. On the other hand, the information contained in the flow table can be obtained using the controller to analyze the attributes of the attacker when a DDoS attack is detected. As a result, the scheme is applicable to SDN.

3 Related Work

In this section, we briefly introduce and analyze the existing DDoS attack detection and defense schemes in SDN.

The scheme based on statistics carries out statistical inference test on data flow, and treating data flow that do not conform to the statistical models as attack data to achieve DDoS attack detection [12,13]. AvantGuard [14] is a scheme to improve the security of DDoS attack detection. It introduces two modules on OF switch. These modules implement attack detection by classifying TCP SYN requests and triggering corresponding actions according to the classification results. However, this scheme can only detect a single type of DDoS attack

(TCP SYN Flood). In 2015, Wang and Jia [15] proposed a scheme to detect DDoS attacks by calculating the IP address entropy of data flow in the network. It can accurately detect DDoS attacks, while it does not provide a defense method for DDoS attacks. In 2017, a scheme to detect unknown attacks via OF switch was proposed by Kalkan et al. [16]. The scheme incorporates intelligence features into the OF switch that enable OF switch to perform independent operations on packets. The act of providing intelligence for OF switch allows the scheme to not only accurately detect known DDoS attacks, but also detect unknown types of DDoS attacks. However, the concept of "capable switch" violates the concept of separate the control plane from the forwarding plane in SDN.

The detection scheme based on machine learning detects DDoS attacks by using a variety of machine learning algorithms to train the detection model [17–19]. In [20], the authors use naive bayes, support vector machine(SVM) algorithms to detect DDoS attacks. The scheme can quickly distinguish the abnormal flow yet the detection accuracy is low. Similarly, SD-Anti-DDoS [21] is proposed to detect DDoS attacks in a fast and efficient manner by Cui et al. The scheme can reduce the load of the controller and OF switch by setting the attack detection trigger to respond to the abnormal attack more quickly. The scheme can detect DDoS attacks that trigger a large number of *packet_in* messages in a short period of time.

In 2018, Cui [22] et al. proposed a time-based detection scheme. The article proposes that the principle of DDoS Attack on SDN Controller is to trigger a large number of packet-in packets, so the attack must result in a sharp drop in the hit rate. As a result, The scheme uses the hit rate of the flow entry as a feature to detect DDoS attack on SDN Controller. This scheme can detect DDoS attack on SDN Controller, but it ignores the spatial feature of the data.

In the above schemes, the feature used is content features basically. Attackers can easily trick detection scheme by adjusting the content of the data packet. In [23], the authors find that the joint entropy can more accurately and flexibly reflect the change of the current data flow state. Since the DDoS attack is inevitably accompanied by the change of feature entropy, it is necessary to use joint entropy to more accurately detect DDoS attack on SDN Controller. In addition, as attack packets often come from different hosts manipulated by attackers, DDoS attacks are often accompanied by an explosion in the number of hosts. Therefore, the number of hosts can also be used as a spatial feature to effectively reflect the current flow changes.

4 The Designed Scheme

In this section, we will describes the details of the proposed scheme. The scheme consists of three modules: Flow process module, Attack detection module and Active defense module. Figure 3 depicts the process of the proposed scheme.

- *Flow Process Module*: This module consists of two-part: Flow collection and Feature process. Flow collection mainly collects packets from the unknown data flow sent by the OF switch to the controller and flow table information for the OF switch. The function of Feature process is to calculate the corresponding feature value according to the extracted packet header information, and forms the sample X.

- *Attack Detection Module*: This module mainly contains a deep neural network model composed of CNN and LSTM, which is responsible for detecting DDoS attack on SDN Controller. The model detects the received the sample X and sends the detection results to Active defense module.
- *Active Defense Module*: According to the detection results sent by Attack detection module and the flow table information, this module will generate the defensive flow entry to defend against DDoS attack on SDN controller.

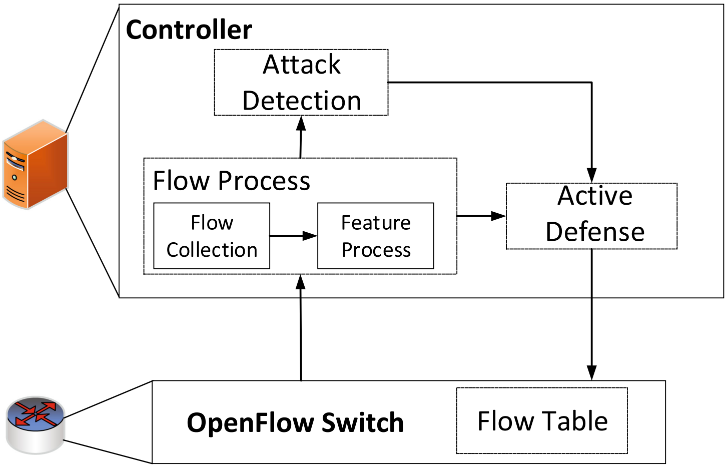

Fig. 3. System architecture of our scheme

4.1 Flow Process Module

Flow process module is embedded in the controller. In SDN, the OF switch send packets that do not match any of the flow entry to the controller. With this mechanism, unknown packets received by the OF switch can be easily collected.

Flow Collection : To detect DDoS attack on SDN Controller, Flow collection collect the different attributes of the unknown packet. The most important attributes of TCP/IP packets headers are: source IP address (IP_{src}), destination IP address (IP_{dst}), source port (P_{src}), destination port (P_{dst}), packet size (PKT_{size}), protocol type (PKT_{type}). These attributes can be represented as a collection:

$$flow = \{IP_{src}, IP_{dst}, P_{src}, P_{dst}, PKT_{size}, PKT_{type}\}$$

When an unknown TCP/IP packet arrives at the controller, Flow collection will collect the property values in the collection *flow* from the packet header. After a period of time, Flow collection consolidates the value received during this period into *Flow*.

In [24], the author compares the time-based and packet-based period determination, and concludes that the packet-based period determination is more effective. So we choose packet-based period determination. Because of this, *Flow* will send to Feature process after Flow collection receives α packets. At the same time, Flow collection will query the OF switch flow table information and sends it to Active defense module. The contents of *Flow* are shown below:

$$Flow = \{flow_1, flow_2, ..., flow_\alpha\}$$

Feature Process : In 1948, the concept of information entropy was presented by Shannon [25]. It can be used to describe the randomness of a random variable. If we consider two independent random variables at the same time. The joint-entropy of random event X and Y is defined as:

$$H(XY) = -\sum_{i=1}^{N}\sum_{j=1}^{M} p_{(}x_iy_j)log_2(p(x_iy_j)) \tag{1}$$

where $p(x_iy_j)$ is the probability of event(X = x_i,Y = y_j),i = 1,2,...,N and j = 1,2,...,M.

Algorithm 1. Framework of ensemble learning for our scheme.

```
Require: The set of receive Flow; The set of attribute pairs $\mathbb{A}$; The number of packets
    received in a period $\alpha$;
Ensure: Sample X
Initialize the set of joint-entropy $\mathbb{JC}_{\mathbb{A}}$;
for $(a_1, a_2)$ in $\mathbb{A}$ do
    Initialize dictionary of counting count_table
    for Each flow in Flow do
        if $(flow.a_1, flow.a_2) \in count\_table$ then
            count_table.add(($flow.a_1, flow.a_2$),1)
        else
            count_table[($flow.a_1, flow.a_2$)] += 1
        end if
    end for
    for Each c in count_table do
        $P = c.key/\alpha$
        $\mathbb{JC}_{\mathbb{A}}.(a_1, a_2) + = P * \log(P)$
    end for
    $\mathbb{JC}_{\mathbb{A}}.(a_1, a_2) / = \log_2(\alpha)$
end for
X = ($\mathbb{JC}_{\mathbb{A}}$, Duration, NUM)
return X
```

In normal traffic, the packet received by the OF switch is random, so the entropy value of normal flow is usually large. In contrast, in a DDoS attack, the entropy of some attributes of attack packet drops dramatically. In the same way, the joint-entropy of the attribute pairs composed of these attributes has the same trend [23]. In addition, when a DDoS attack on SDN Controller occurs, attack packets received by the OF switch increase significantly and the number of hosts corresponding to the flow will increase. At the same time, the duration of the flow containing attack packets will also be shortened. These features can be extracted to reflect changes in the spatial characteristics of the flow.

The attributes in the collection *flow* constitute the attribute pair. There attribute pairs form a collection $\mathbb{A}$. The contents of collection $\mathbb{A}$ are shown below:

$$\mathbb{A} = \{(IP_{src}, IP_{dst}), (IP_{src}, IP_{dst}), ..., (PKT_{size}, PKT_{type})\}$$

We use A_i represents the i^{th} element of the collection $\mathbb{A}$, JC_{A_i} represents the corresponding joint-entropy value of A_i, respectively. For example, $A_1 = (IP_{src}, IP_{dst})$. JC_{A_1} represents the joint-entropy of attribute pair consisting of source IP address and destination IP address. The process of Feature process is shown in algorithm 1.

Finally, Feature process generates the sample X, which represents the characteristics of the flow. *Duration* and *NUM* represents the duration of *Flow* and the number of hosts in the *Flow*, respectively.

4.2 Attack Detection Module

In this module, we construct a deep neural network model to detect DDoS attack on SDN Controller. The view of the model is shown in Fig. 4. We choose the CNN-LSTM as the core of the model. Since CNN has been proved to be able to extract the spatial feature of data efficiently. LSTM is not only good at processing sequential data, but also avoids the gradient disappearance during training [26,27].

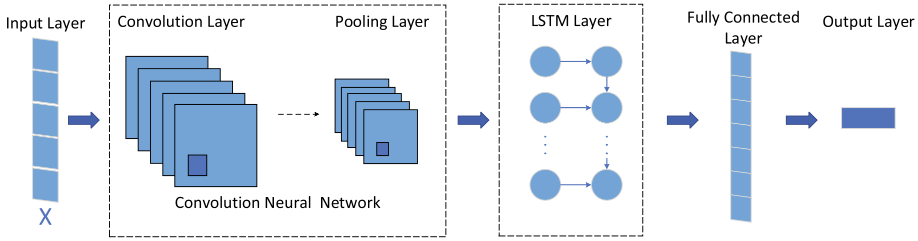

Fig. 4. Overall CNN-LSTM module

The model consists of input layer, convolution layer, pooling layer, LSTM layer, full connection layer and output layer, which are combined according to linear structure. The data of the input layer is the sample X. The sample first is processed by the convolutional neural network composed of the convolution layer and pooling layer. The convolution layer extracts features of high dimensional by performing convolution operation on the sample X, while the pooling layer compresses and reduces dimensions of features of high dimensional extracted by the convolution layer to simplify the complexity of network computation. In this model, we set the convolutional kernel of the convolutional layer as 3 and the step size as 1. In order to speed up the training progress of the model, we use the ReLU function as the activation function of the convolutional layer.

$$ReLU(x) = max(0, x) \tag{2}$$

After the convolution network processing, the output of the high-dimensional spatial feature with the sample X will be input into the LSTM layer for further processing. The LSTM layer consists of several layers of LSTM units. With the introduction of the concept of cell, LSTM can effectively recognize the implied temporal relationship between large sequential data. Therefore, the output processed by LSTM layer has the spatial-temporal characteristics of sample X.

Finally, the output of the LSTM layer, after being processed by the fully connected layer, will generate the final output through an output layer to determine whether the current network is under DDoS attack. For the output layer classifier, we use the value of the Softmax function as the output.

$$Softmax(x_i) = \frac{e^{x_i}}{\sum_j e^{x_j}} \tag{3}$$

All in all, the sample X obtained will be brought into the trained model, and the model will classify the sample X according to the feature it has. The detection results will send to Active Defense Module.

4.3 Active Defense Module

After receiving the results from Attack detection module. Active defense module will select suspicious flow entries to determine the attacker's attribute value, and generate defensive flow entry to defend against DDoS attacks.

As mentioned above, most DDoS attacks on SDN Controller leverage the OpenFlow processing mechanism. There DDoS attacks consume controller resources quickly by generating as many attack packets as possible. Suspicious flow entries are those generated by attack packets. Since they are generated by attack packets, this means that suspicious flow entries typically match to fewer packets and have a longer lifetime than normal flow entries. At the same time, the traffic matching of these flow entries often has a high asymmetry characteristic.

When the number of matched packets of any flow entry is lower than the mean A_p, this module will calculate the duration and asymmetric flow rate of the flow entry, and compare with the mean values of A_s and A_f respectively. If all of these values are above the mean values, this flow entry is considered as

Table 1. Defensive flow entry

Priority	Match field	Counter	Instructions
Minimal	$IP_{Src} = A_{ack}$	...	Actions = Drop

a suspicious flow entry. Through the suspicious flow entry, defensive flow entry will generated. An example of a defensive flow entry is shown in Table 1.

The function of this flow entry is to drop all packets sent from the A_{ack} address. A_{ack} is believed to be the IP address of the attacker. These attributes can be obtained from suspicious flow entry that are generated by the attacker.

Although the defensive flow entry can effectively block the attack packets from the attacker, sometimes it may also block legitimate packets sent by normal users. So this module will remove the defensive flow entry immediately after the DDoS attack is over to mitigate its impact. We believe that the attack still occurs when a large number of unknown packets are sent to the victim. Therefore, the defensive flow entry will match a large number of packets. By calculating the ratio of the number of packets that matched the defensive flow entry to the normal flow entry, we can determine whether the attack is over. The calculation formula is as follows:

$$N_n = \frac{1}{C_n} \sum_{i \in T_n} FlowCount_i \tag{4}$$

$$N_d = \frac{1}{C_d} \sum_{j \in T_d} FlowCount_j \tag{5}$$

The T_n, T_d represents the set of normal flow entry and defensive flow entry in the OF switch, and the C_n, C_d represents the number of flow entries of sets T_n and T_d, respectively. Finally, we will get the ratio of the number of matched packets between the normal flow entries and the defensive flow entries. And if the result satisfies Eq. 6, we judge that the victim is no longer under the DDoS attack, and remove the defensive flow entry.

$$\frac{N_d}{N_n} < \lambda \tag{6}$$

5 Experiments and Evaluation

5.1 Experiment

DARPA1999 data set [28] is selected to verify the effectiveness of the scheme. By making comparative experiments, we found that when set $\alpha = 100$, that is, one hundred packets are collected as a processing period, the effect of the whole scheme is the best. In addition, we set up several groups of control tests, and

Table 2. Threshold test

Value of the λ	Precision	Misjudgment rate
5	1.00	0.43
7	0.98	0.31
10	0.93	0.14
15	0.81	0.06
20	0.43	0.03

select different values of λ for each group of tests to determine the most suitable value. The experimental results are shown in Table 2. Finally, we found that when $\lambda = 10$, defensive flow entry can correctly block the flow of data from attackers and have little effect on normal data. The parameters used in the experiment and their meanings are shown in Table 3.

Table 3. System parameters

Term	Explanation
$flow$	A collection of data packet header properties
$Flow$	A collection of data collected over a period
$\mathbb{A}$	A collection of attribute pairs formed by the combination of collected data packet header attributes
$\mathbb{JC}_{\mathbb{A}}$	The set of joint entropy corresponding to the attribute pairs in set $\mathbb{A}$
$count\_table$	A dictionary that stores and counts the value of a pair of features in a period and the number of times the value
α	The number of data packets collected in a period ($\alpha = 100$ in our experiment)
A_p	The average of the number of packets matched by the flow entries
A_s	The average duration of flow entries
A_f	The average flow rate of flow entries asymmetric flow
λ	Threshold for the ratio of the number of matched packets between the defensive and normal flow entry

We use Mininet [29] as the network simulator, which can help us build a network topology similar to the real environment. We also use OpenDayLight, an open source controller, as SDN controller. As for verifying that the scheme can effectively block the traffic of DDoS attack, we simulated several DDoS attacks. The types and duration of simulated DDoS attacks are shown in Table 4. And for the southbound interface protocol, we chose OpenFlow1.3, since it is the most commonly used protocol in the industry. The simulation runs on an 8GRAM, Intel Core i5-4590 3.30GHz CPU with Ubuntu 14.04OS.

5.2 Performance Metrics

We conduct comparative experiments with the scheme proposed in [22]. Because the scheme [22] takes into account the time feature of a DDoS attack and achieves good results in the detection and defense of DDoS Attack on SDN Controller.

Table 4. Attack description

Attack Type	Principal	Start Time(s)	End Time(s)
Portsweep	Attacker sends some packet to every port of the target network to determine which host is available to attack	100	160
Ipsweep	Attacker sends some packet to every host of the target network to determine which host is available to attack	100	160
smurf	Attacker sends massive ICMP packets with forged source IP address to the target host, then the target host replies to all nonexistent source Hosts, and become too busy to handle other legitimate packets	340	400
neptune	Attacker sends massive SYN packets with different ports to target host, then the target host replies every SYN packet and waits, finally All the ports of target host are occupied	460	490

We use the ACC(Accuracy), P(Precision) and R(Recall) for evaluating the parameters of the detection efficiency. The parameters are defined as follows:

- *Accuracy(ACC)*:the proportion of data samples that are correctly classified to the total data samples.

$$ACC = \frac{TP + TN}{TP + TN + FP + FN}$$

Table 5. Accuracy, precision and recall

	Accuracy	Precision	Recall
LSTM	0.922	0.948	0.936
CNN	0.907	0.939	0.947
BPNN(Proposed by [22])	0.827	0.921	0.91
CNN-LSTM(Our Proposed)	**0.943**	**0.988**	**0.954**

- *Precision(P)*:the proportion of true attack samples in the attack samples determined by the algorithm.

$$P = \frac{TP}{TP + TN}$$

- *Recall(R)*:the proportion of the attack samples that have been correctly determined to the total attack samples.

$$R = \frac{TP}{TP + FN}$$

The experimental results are shown in Table 5. This proposed scheme can detect DDoS attack on SDN Controller more efficiently, because it can extract the spatial and temporal characteristics of the data flow.

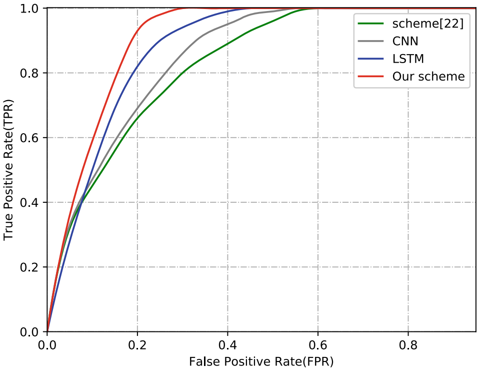

Fig. 5. ROC curve comparison for different algorithms

TP(True Positive) represents the number of samples that the detection algorithm correctly determines to be attacked. TN(True Negative) represents the number of normal samples that are determined correctly. In that vein, FP(False Positive) is the number of normal samples that are incorrectly identified as attack samples. FN(False Negative) represents the number of attack samples that are incorrectly determined as normal samples.

The Receiver Operating Characteristic(ROC)curve, as a standard measure of classifier classification, can reflect the performance of the classifier. As shown in Fig. 5, the scheme proposed is superior in detecting DDoS attack on SDN Controller.

The CPU consumption rate of the victim host is shown in Fig. 6. Since this scheme can quickly detect the attack and send out the defensive flow entry to block the attack data, so that the CPU utilization of the victim service can always maintain the normal state. Compared with the scheme [22], the defensive flow entry issued by our method can regulate network traffic in a more granular way instead of directly cutting off the traffic sent to the victim's IP address, so the CPU consumption rate of this scheme is lower.

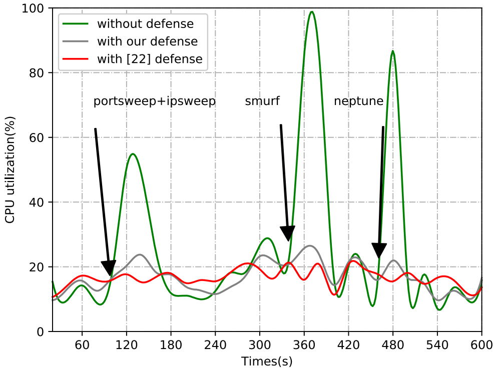

Fig. 6. System CPU utilization

6 Conclusion

This paper proposes a scheme to detect and defense DDoS attack on SDN Controller based on spatial-temporal feature. By extracting the spatial and temporal characteristics of the data flow, the scheme can accurately detect DDoS attack on SDN Controller. The defense module of the scheme generates the defensive flow entry through the lightweight calculation to carry on the fine-grained regulation to the data flow. The experiment is operated on the DARPA1999 data set. The experimental results show that the proposed scheme could detect DDoS attack on SDN Controller more accurately.

References

1. Lopes, F.A., et al.: A software engineering perspective on sdn programmability. IEEE Commun. Surveys Tuts. **18**(2), 1255–1272 (2016). https://doi.org/10.1109/COMST.2015.2501026
2. Scotthayward, S., Sriram, N., Sakir, S.: A survey of security in software defined networks. IEEE Commun. Surveys Tuts. **18**(1), 623–654 (2016). https://doi.org/10.1109/COMST.2015.2453114
3. Swami, R., Mayank, D., Virender, R.: Software-defined networking-based ddos defense mechanisms. ACM Comput. Surveys **52**(2), 1–36 (2019). https://doi.org/10.1145/3301614
4. Yan, Q., et al.: Software-defined networking (sdn) and distributed denial of service (ddos) attacks in cloud computing environments: a survey, some research issues, and challenges. IEEE Commun. Surveys Tuts. **18**(1), 602–622 (2016). https://doi.org/10.1109/COMST.2015.2487361
5. Praseed, A., Santhi, T.P.: DDOS attacks at the application layer: challenges and research perspectives for safeguarding web applications. IEEE Commun. Surveys Tuts. **21**(1), 661–685 (2019). https://doi.org/10.1109/COMST.2018.2870658
6. Han, B., et al.: OverWatch: a cross-plane DDOS attack defense framework with collaborative intelligence in SDN. Secur. Commun. Netw. 1–15 (2018) https://doi.org/10.1155/2018/9649643

7. Wang, Y., et al.: SGS: safe-guard scheme for protecting control plane against ddos attacks in software-defined networking. IEEE Access 34699–34710 (2019) https://doi.org/10.1109/ACCESS.2019.2895092
8. Kalkan, K., Gurkan, G., Fatih, A.: Defense mechanisms against ddos attacks in sdn environment. IEEE Commun. Mag. **55**(9), 175–179 (2017). https://doi.org/10.1109/MCOM.2017.1600970
9. Kumar, K., Joshi, R. C., Singh, K.: A distributed approach using entropy to detect DDoS attacks in ISP domain. In: International Conference on Signal Processing, pp. 331–337. (2007). https://doi.org/10.1109/ICSCN.2007.350758
10. Barki, L., et al.: Detection of distributed denial of service attacks in software defined networks. In: Advances in Computing and Communications, pp. 2576–2581 (2016). https://doi.org/10.1109/ICACCI.2016.7732445
11. Mckeown, Nick, et al. "OpenFlow: enabling innovation in campus networks." acm special interest group on data communication (2008): 69–74. https://doi.org/10.1145/1355734.1355746
12. Xu, Y., Yong, L.: DDoS attack detection under SDN context. In: IEEE International Conference Computer And Communications, pp. 1–9 (2016). https://doi.org/10.1109/INFOCOM.2016.752450
13. Kumar, P., et al.: SAFETY: early detection and mitigation of TCP SYN flood utilizing entropy in SDN. IEEE Trans. Netw. Service Manag. **15**(4), 1545–1559 (2018). https://doi.org/10.1109/TNSM.2018.2861741
14. Shin, S., et al.: AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks. In: Computer and Communications Security, pp. 413–424 (2013) https://doi.org/10.1145/2508859.2516684
15. Wang, R., Zhiping, J., Lei, J.: An entropy-based distributed ddos detection mechanism in software-defined networking. In: Trust, Security and Privacy in Computing and Communications, pp. 310–317 (2015) https://doi.org/10.1109/Trustcom.2015.389
16. Kalkan, K., Gurkan, G., Fatih, A.: SDNScore: A statistical defense mechanism against DDoS attacks in SDN environment. In: International Symposium on Computers and Communications, pp. 669–675 (2017). https://doi.org/10.1109/ISCC.2017.8024605
17. Xie, J., et al.: A survey of machine learning techniques applied to software defined networking (sdn): research issues and challenges. IEEE Commun. Surveys Tuts. **21**(1), 393–430 (2019). https://doi.org/10.1109/COMST.2018.2866942
18. Latah, M., Levent, T.: Artificial intelligence enabled software-defined networking: a comprehensive overview. IET networks **8**(2), 79–99 (2019). https://doi.org/10.1049/iet-net.2018.5082
19. Dayal, N., et al.: Research trends in security and ddos in sdn. Secur. Commun. Netw. **9**(18), 6386–6411 (2016). https://doi.org/10.1002/sec.1759
20. Deepa, S., Deepa, L.: Detection of ddos attack on sdn control plane using hybrid machine learning techniques. In: International Conference on Smart Systems and Inventive Technology, pp. 299–303 (2018) https://doi.org/10.1109/ICSSIT.2018.8748836
21. Cui, Y., et al.: SD-Anti-DDoS: fast and efficient ddos defense in software-defined networks. J. Netw. Comput. Appl. **68**, 65–79 (2016). https://doi.org/10.1016/j.jnca.2016.04.005
22. Cui, J., He, J., Xu, Y., Zhong, H.: TDDAD: time-based detection and defense scheme against ddos attack on sdn controller. In: Susilo, W., Yang, G. (eds.) ACISP 2018. LNCS, vol. 10946, pp. 649–665. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-93638-3_37

23. Mao, J., Weijun, D., Fuke, S.: DDoS flooding attack detection based on joint-entropy with multiple traffic features. In: trust security and privacy in computing and communications, pp. 237–243 (2018). https://doi.org/10.1109/TrustCom/BigDataSE.2018.00045
24. Kim, Y., et al.: Packetscore: statistics-based overload control against distributed denial-of-service attacks. In: International Conference on Computer Communications, pp. 2594–2604 (2004). https://doi.org/10.1109/INFCOM.2004.1354679
25. Shannon, C.E.: Prediction and entropy of printed English. Bell Syst. Tech. J. **30**(1), 50–64 (1951). https://doi.org/10.1002/j.1538-7305.1951.tb01366.x
26. Cui, J., et al.: Comparative study of CNN and RNN for deep learning based intrusion detection system. In: International Conference on Cloud Computing, pp. 159–170 (2018). https://doi.org/10.1007/978-3-030-00018-9_15
27. Zhai, S., et al.: Deep structured energy based models for anomaly detection. In: International Conference on Machine Learning, pp. 1100–1109 (2016)
28. MITLincolnLaboratory:DARPA 1999 Intrusion Detection Data Set. https://www.LL.mit.edu/ideval/docs/attackDB.html
29. Mininet. http://mininet.org/