International Core Journal of Engineering Volume 7 Issue 6, 2021
ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

DDoS Attack and Flash Event Detection Method based on One-dimensional Convolutional Neural Network in SDN Network

Jiawei Yang[1,a], Anjun Song[1]

[1]Shanghai Maritime University, Shanghai 201306, China.

[a]yang_ow@126.com

Abstract

Nowadays, distributed denial of service attacks have become an important threat to Internet security. In software defined networking (SDN), due to its centralized management and control characteristics, it is easy to become the target of DDoS attacks, leading to nodes in the network. Invalidate. However, as the Flash Event (FE) event is known to more and more professionals, due to the many common characteristics of the two, it is very necessary to effectively distinguish the two events. This paper proposes a method for distinguishing and detecting DDoS attack traffic and flash traffic based on a one-dimensional convolutional neural network. This method collects the flow table items in the switch, and then integrates the data into 7 important features, including the calculation of ip entropy based on φ-entropy, etc., and uses a one-dimensional convolutional neural network model to perform the collected data Training and classification, and finally realize the detection of DDoS attacks, flash flood events and ordinary traffic. The experimental results show that compared with the commonly used classification algorithms, the detection method proposed in this paper has improved accuracy, recall, precision and F1 score, which proves the effectiveness of the method.

Keywords

Software Definition Network; DDoS Attack; Flash Event; One-dimensional Convolution Neural Network; Attack Detection.

1. Introduction

With the continuous development of the network, network attacks are constantly expanding and evolving. Denial of attack (DoS) is a common attack method, which can make users and enterprise servers busy processing useless requests, so that they can not respond to normal requests. Distributed Denial of Attack (DDoS) is an attack method based on a large number of puppet hosts carrying out DoS attacks on the target host at the same time. This kind of behavior that seems to be continuously sending requests from all over the world can consume the server's processor and bandwidth to a large extent, resulting in server downtime and service offline. The scale of this attack method is now becoming larger and larger. According to a Kaspersky Lab report[1], compared to the fourth quarter of 2019, the number of DDoS attacks in the first quarter of 2020 has doubled. An increase of 80% compared to the same period last year. At the same time, according to the report of Amazon cloud service AWS[2], a record-breaking DDoS attack was detected in February 2020, and its peak traffic reached 2.3 terabits per second. These data indicate that DDoS attacks are at the frequency of attacks. There is an upward trend in intensity.

In addition to DDoS, there is currently a network event that is being valued by more and more researchers. It is called Flash Event or Flash Crowd. Its essence is that a large number of users initiate requests to the server at the same time because of an event, which leads to a sudden increase in server

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

pressure, which may seriously cause downtime [3]. This situation generally occurs in, for example, important World Cup matches, presidential elections, and public events. Many of its features are very similar to DDoS, for example, the server suddenly receives a large number of data packets from different source IPs. Although this kind of event may also cause the server to crash, these requests are legitimate and need to be responded to, not filtered out.

Software defined network (SDN, software define network) is a new type of network technology. The difference between it and the traditional network structure is that its entire forwarding strategy is determined by an independent entity called a controller, so the decision plane of SDN is Independent of the data plane, so as to achieve centralized management of the network. In the SDN network, when an unfamiliar data packet comes in, the switch will first query its own static flow table. If there is no corresponding flow table to guide its forwarding, the switch will send a packet-in message to the controller, and the controller will receive it. After the packet-in message, the flow table will be sent to the controller according to the network structure to instruct it to forward the flow table. This working principle is extremely easy to be exploited by DDoS. Attackers continue to send data packets with fake ips to servers in the SDN network through a large number of puppet hosts. Switches continue to send data packets because they cannot find the corresponding flow table from the static flow table. The controller sends packet-in messages. In the face of huge attack traffic, the processing resources of the controller will be quickly exhausted. At the same time, the switch will be occupied by a large amount of space by the continuously generated flow table, thus losing the forwarding ability.

At present, many studies have focused on distinguishing the malicious traffic generated by DDoS from normal traffic. However, because the performance of Flash Event will be similar to DDoS, the Flash Event traffic that should have been responded to is rejected. Therefore, it is very necessary to distinguish between DDoS attacks and Flash Event on SDN networks.

2. Related work

Since a large number of unfamiliar source Ips will flood into the network during a DdoS attack, it will inevitably increase the chaos of the entire network environment. Entropy, as a measure of randomness and consistency of a system, can just be used as a tool to detect whether the network has received DdoS Attacks. Sachdeva et al. [4] calculated cluster entropy and source IP entropy to distinguish distributed denial of service attacks and Flash Event, and verified their proposed method through simulation in NS2 and simulation in DETER test bed. In the experiment, the CAIDA dataset was used to represent DDoS attacks and the FIFA World Cup dataset was used as the dataset to verify their method. Behal et al. [5] studied the applicability of generalized entropy (GE) and generalized information distance (GID) indicators based on information theory in detecting different types of DdoS attacks, and compared the results of generalized entropy and generalized information distance measurement with the Shannon entropy sum Comparing other popular information difference measures, the final result shows that the generalized entropy and generalized information distance indicators perform well compared with other indicators, and the False Positive Rate (FPR) is reduced.

When the behavior of detecting an attack is regarded as a classification problem, since its network characteristics can be integrated into data, traditional learning can also be used to distinguish malicious traffic from traditional traffic. Wenti Jiang et al. [6] proposed a multi-type DdoS attack and flash traffic detection method based on flow characteristics in SDN. The flow characteristics generated in DdoS and FE events are collected and distinguished by KNN algorithm, the highest F1 score was 0.947, and the detection rate of Flash Event was 0.903. Lingjiao Wang et al. [11] proposed a DdoS attack detection algorithm based on Support Vector Machine (SVM)—RF-SVM (Random Forest-SVM). At first, this method selects the associated six-dimensional features by combining the data packet header information, then uses the random forest to calculate the feature weights and filters the features to obtain an optimal feature subset, and finally uses the SVM algorithm to detect DdoS attacks. The experimental results in the same scene show that the RF-SVM algorithm has higher detection, recall and F1 score than SVM algorithm and RF algorithm.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

Neural networks can also be used to deal with such problems because of their ability to automatically extract features from data. Chuanhuang Li et al. [7] combined the two machine learning models of DCNN and DASE as a recognition model, collected fields in the flow table and constructed additional features to send to the model for training, and the accuracy of distinguishing between attack traffic and ordinary traffic reached 97%. Xiaorui Wang et al. [8] proposed a DdoS attack detection method based on BP neural network in SDN environment. By analyzing the changes of six related feature values, the BP neural network algorithm is used to classify training samples to realize the detection of DdoS attacks.

The above-mentioned methods have more or less problems of incomplete detection range or low accuracy, so this paper proposes a DdoS and FC event detection method based on φ-entropy and one-dimensional convolutional neural network model. On the basis of ensuring simultaneous detection of DdoS attacks and FC events, it also has better performance in accuracy, recall, precision and F1 scores.

3. Feature construction

So far, distinguishing between DDoS attacks and FE incidents is an inevitable problem for those engaged in network security. DDoS attacks and FE incidents have many common characteristics, such as the sudden increase in traffic entering the network, but there are still There are many parameter differences. Behal et al. [9] summarized the feature differences between DDoS and FE events in traditional networks. However, in SDN networks, the features that the two can extract are slightly different from those in traditional networks. 1 shows the main difference between the two in the SDN network.

Table 1. The main differences between DDoS attacks and Flash Event

NO.	Parameter	DDoS	Flash Event
1	Flow similarity	high	low
2	Server response	low	high
3	Continuous connection time	low	high
4	Request type	GET	GET/POST
5	Number of requests per ip	low	high
6	Destination ip distribution	Uniform	Similar to the two-eight law

3.1 Introduction to φ-Entropy

Based on the existing Shannon entropy and generalized entropy, Behal et al. proposed a new measurement information system φ-entropy [4], the calculation formula is shown in formula 1:

$$H_\alpha(X) = -\frac{1}{\sinh(\alpha)}\left(\sum_{i=1}^{n} P_i \sinh(\alpha \log_2 P_i)\right) \quad (1)$$

In the formula, P_i represents the probability of occurrence of event X, and α is the key parameter to adjust the sensitivity of the entire formula. When the degree of randomness of the two systems is not much different, the traditional information entropy value is also not much different, so when using φ-entropy, the different sensitivity brought by different α can effectively distinguish two systems with a small difference. Fig. 1 shows the performance of traditional information entropy and φ-entropy in DDoS attack data and FE events. It can be clearly seen that φ-entropy performs better and distinguishes the two systems more effectively.

3.2 Flow feature collection and integration

The characteristic data used in the experiment is collected through the openflow protocol. In the openflow protocol, the flow table is a guide for the switch to forward data packets. Each flow table is composed of header field, counters, and actions [10], when a new data packet arrives at the switch, the switch does not have a matching flow table to guide its forwarding, so the switch will integrate the data packet into the packet-in message and send the message to the controller. The controller

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

passes After processing, the switch is instructed to write the flow table through the flow_mod message. At this time, the switch knows how to process the data packet. The switch periodically sends ofp_flow_stats_request messages to the controller, and the static flow table information in the corresponding switch can be collected through sudo ovs-ofctl dump-flows s1. Fig. 2 shows a sample diagram of the flow table available for collection.

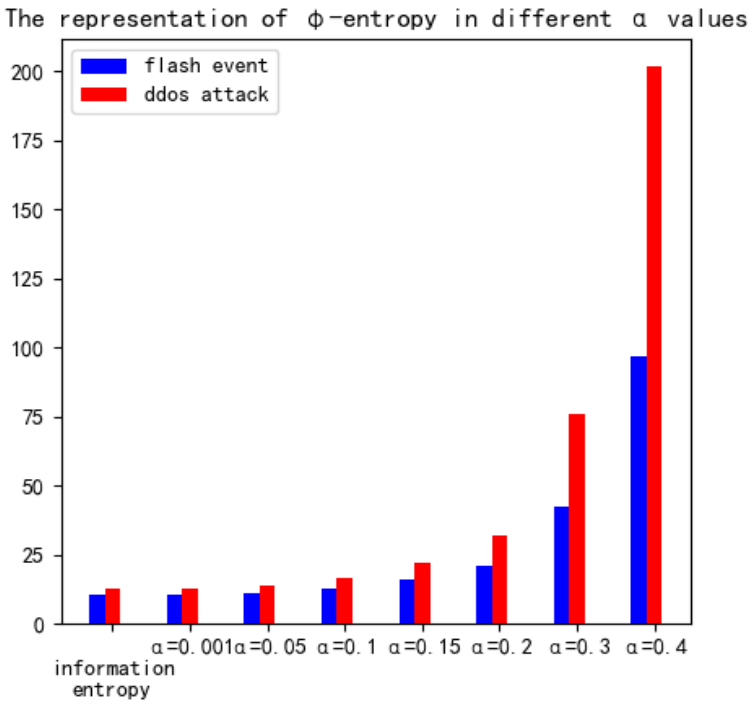

Fig. 1 The performance graph of traditional information entropy and φ-entropy under different α

```
NXST_FLOW reply (xid=0x4):
 cookie=0x2005f7c6000000, duration=4.878s, table=0, n_packets=0, n_bytes=0, idle_timeout=5, idle_age=4, priority=1,ar
p,in_port=2,dl_src=d2:9c:5e:8a:99:e7,dl_dst=fa:99:ab:76:39:89 actions=output:3
 cookie=0x2005f7c7000000, duration=4.874s, table=0, n_packets=3, n_bytes=294, idle_timeout=5, idle_age=2, priority=1,
ip,in_port=3,dl_src=fa:99:ab:76:39:89,dl_dst=d2:9c:5e:8a:99:e7,nw_src=10.1.1.1,nw_dst=10.10.10.2 actions=output:2
 cookie=0x2005f7c8000000, duration=4.873s, table=0, n_packets=2, n_bytes=196, idle_timeout=5, idle_age=2, priority=1,
ip,in_port=2,dl_src=d2:9c:5e:8a:99:e7,dl_dst=fa:99:ab:76:39:89,nw_src=10.10.10.2,nw_dst=10.1.1.1 actions=output:3
 cookie=0x0, duration=14.222s, table=0, n_packets=118, n_bytes=12254, idle_age=0, priority=0 actions=CONTROLLER:65535
```

Fig. 2 Sample diagrams of flow tables available for collection

When the network receives an attack or an FE event occurs, the network will fluctuate. A large number of IPs will flood into it. Some servers will receive the impact of traffic, and the flow entry will increase rapidly. The data packet size will show different characteristics. Therefore, based on the above analysis, relevant data is extracted from the flow table and integrated into one-dimensional features when different events occur, including the following 7 items.

3.2.1 Average bytes of data packet

Since DDoS attacks are initiated by the puppet host, most of the data packets sent are of fixed size. For example, in the default state, the packet size of udp attacks displayed in the flow table is mostly 92B, icmp attacks are mostly 142B, and tcp attacks. Most are 204B, and the packet size of ping communication between hosts is 42B. When an FE event occurs, since each request is a normal user behind it, it is inevitable that the packet size of the entire event request is inconsistent to the same degree. Therefore, this data can be used as one of the basis for distinction. The specific calculation formula is shown in formula 2:

$$Byte_{avg} = \frac{\sum_{i=0}^{flow\_count} n\_bytes}{flow\_count} \quad (2)$$

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

In the formula, n_bytes represents the size of the packet bytes in each flow table, and flowcount represents the number of flow table in the collected data.

3.2.2 Average age of flow table

The duration data in the flow table indicates the time since the flow table was created, which is the age of the flow table. When a large number of data packets enter the network, a large number of flow tables will be created, and the duration in the flow table data extracted each time will be too small. Therefore, the characteristics of the current network state can also be obtained by analyzing the duration data. The specific calculation formula is as follows As shown in formula 3:

$$Age_{avg} = \frac{\sum_{i=0}^{flow\_count} duration}{flow\_count} \quad (3)$$

In the formula, duration represents the duration data in the flow table, and flow_count represents the number of flow tables in the collected data.

3.2.3 φ-entropy of source ip

When DDoS attacks and FE events occur, it is not easy to effectively distinguish DDoS from FE by simply calculating the number of new source ips. Therefore, the entropy calculation method based on φ-entropy is used to reflect the degree of dispersion of the current network state. The specific calculation formula As shown in formula 4:

$$Src\_ip_{\varphi} = -\frac{1}{\sinh(\alpha)}\left(\sum_{i=1}^{n} P_i \sinh(\alpha \log_2 P_i)\right) \quad (4)$$

In the formula,α is a constant for adjusting the sensitivity of φ-entropy, P_i represents the probability that the i-th ip appears in the source ip pool in the collected flow table data, and n represents the length of the source ip pool in the collected flow table data.

3.2.4 φ-entropy of the destination ip

Because the purpose of DDoS attacks is to paralyze the target server, its attack targets are relatively concentrated. Although most FE events access the same content on the server at the same time, their distribution is similar to the law of 8-2, so through the φ- The entropy value of the entropy is analyzed, and the specific calculation formula for the access target in the current network can be obtained as shown in formula 5:

$$Dst\_ip_{\varphi} = -\frac{1}{\sinh(\alpha)}\left(\sum_{i=1}^{n} Q_i \sinh\left(\alpha \log_2 Q_i\right)\right) \quad (5)$$

In the formula,α is a constant for adjusting the sensitivity of φ-entropy, Q_i represents the probability that the i-th ip appears in the destination ip pool in the collected flow table data, and n represents the length of the destination ip pool in the collected flow table data.

3.2.5 Flow table growth per unit time

Since a new flow table is generated when a new request arrives on the network, and the flow table generation rate is different between normal and abnormal network conditions, the growth of the new flow table under normal conditions shows a relatively stable trend, when DDoS attacks and FE incidents occur It will increase rapidly, so the increase number of the statistical unit time flow table can be calculated according to the current network status. The specific calculation formula is shown in formula 6:

$$Flow_{increase} = \frac{flow\_count}{\Delta T} \quad (6)$$

In the formula,flow_count represents the number of flow tables in the collected data, and ΔT represents a certain time interval.

3.2.6 Average number of data packets in the flow table

In DDoS attacks, sometimes in order to increase the attack efficiency, the data packets are fragmented and then sent out. This kind of data packets use random ips in disguise, so the number of data packets corresponding to each fake ip is small, and the normal request is The corresponding ip and data

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

packets will show relatively more performance. The specific calculation formula is shown in formula 7:

$$Packet_{avg} = \frac{\sum_{i=0}^{flow\_count} n\_packet}{flow\_count} \quad (7)$$

In the formula, n_packet is the number of data packets in the flow table, and flow_count is the number of collected flow tables.

3.2.7 Proportion of pair flow

pair flow refers to that the two flow tables A and B meet the same network protocol, and the destination IP address and source IP address of the flow table A are the source IP address and destination address of the flow table B. Conversely, when a flow table cannot find the corresponding convection, it is called a single flow. When a normal network request occurs, two flow tables will be left in the switch. However, in DDoS attacks and FE incidents, due to the huge amount of traffic, the server cannot handle all of them, so it will only be generated in the switch A flow table, so the current network state can also be obtained by analyzing the number of convections in the flow table data. The specific calculation formula is shown in formula 8:

$$Proportion_{pair\_flow} = \frac{n\_pair\_flow}{n\_pair\_flow + n\_single\_flow} \quad (8)$$

In the formula, n_pair_flow is the number of convections satisfying the convection law, and n_single_flow is the number of single flows satisfying the single flow law.

4. DDoS detection based on one-dimensional convolutional neural network

4.1 Attack detection architecture

The DDoS attack detection architecture in this article is shown in Fig. 3 Through regular flow table collection on the switch, 7-dimensional feature values are extracted and sent to the model for classification judgment. If the prediction result is a DDoS attack, the corresponding defense measures are implemented, as follows Send a specific flow table and other operations. If it is predicted to be normal traffic, no operation is performed, and data packets are forwarded normally. If traffic is predicted to be an FE event, operations such as load balancing are performed to ensure that the user experience is not affected as much as possible .

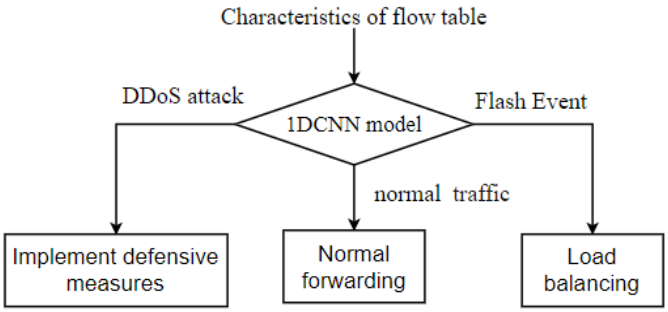

Fig. 3 DDoS attack detection architecture diagram

4.2 Introduction and model of one-dimensional convolutional neural network

Since the convolutional neural network was proposed, it has been widely used in image classification and other fields by virtue of its local connection and weight sharing characteristics. In this field, the input of the network, the convolution kernel, and the internal structure of the network are all two-dimensional. In this topic, the experimental data is one-dimensional data composed of features, which is more suitable for processing signal processing or time series processing. Dimensional Convolutional Neural Network.

The one-dimensional convolutional neural network designed in this experiment is shown in Fig. 4:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

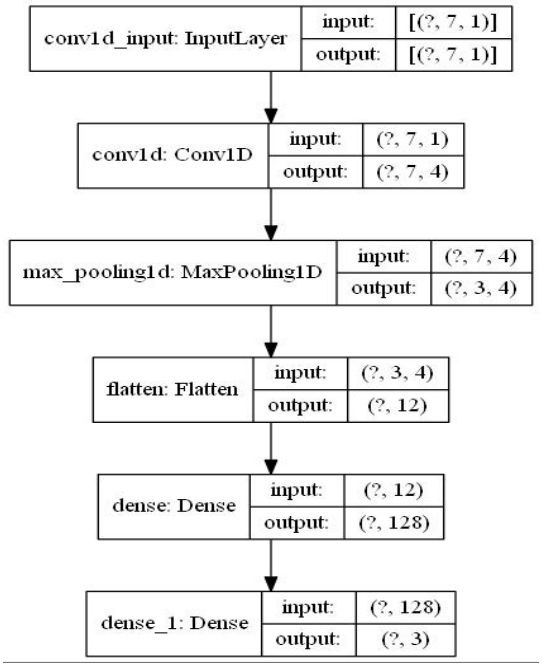

Fig. 4 One-dimensional convolutional neural network structure diagram

The entire network consists of an input layer, a convolutional layer, a pooling layer, a flat layer, a fully connected layer, and an output layer. The function of the convolutional layer is to perform feature extraction on the input one-dimensional data. Each convolution layer is composed of multiple convolution kernels. The convolution verification is a weight matrix. The matrix is convolved with the input data, and then added with the paranoia to obtain the convolution result. The specific calculation process is as follows formula 9 shows:

$$X_m^{out}=f\left(\sum x_m^i * w_m^i+b_m^i\right) \tag{9}$$

In the formula, X_m^i is the input data of the i-th convolution area of the m-th layer, w_m^i is the weight parameter of the i-th convolution kernel of the m-th layer, and b_m^i is the paranoid parameter corresponding to the m-th layer, f is the activation function, and X_m^{out} is the output result after the m-th layer convolution.

The pooling layer usually appears after the convolutional layer, and is generally divided into a maximum pooling layer and an average pooling layer. Like the convolutional layer, it has a spatial window. By continuously sliding the window, the data is cut and extracted. The corresponding features can reduce the number of parameters in the network, save resources, and effectively prevent overfitting. The schematic diagram of the maximum pooling layer used in the model is shown in Fig. 5, and the formula is shown in formula 10:

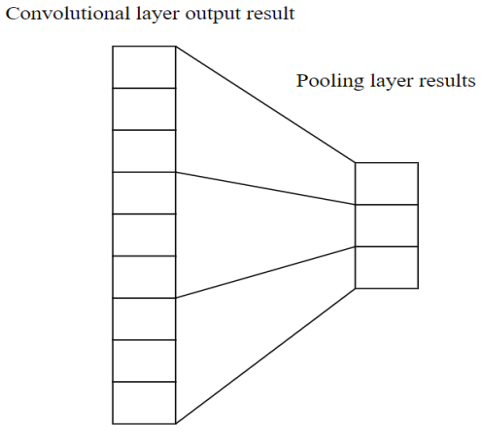

Fig. 5 Schematic diagram of the maximum pooling layer

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

$$T_i = \max(p_i) \tag{10}$$

In the formula, P_i is the output result of the convolutional layer whose size is the pooling size.

The fully connected layer plays a role in the classification of the entire convolutional neural network. By integrating the features extracted by the pooling layer and the convolutional layer, they are mapped to the sample space fully connected layer. Its essence is through 1*1 convolution The kernel performs convolution operation. The schematic diagram of the fully connected layer is shown in Fig. 6:

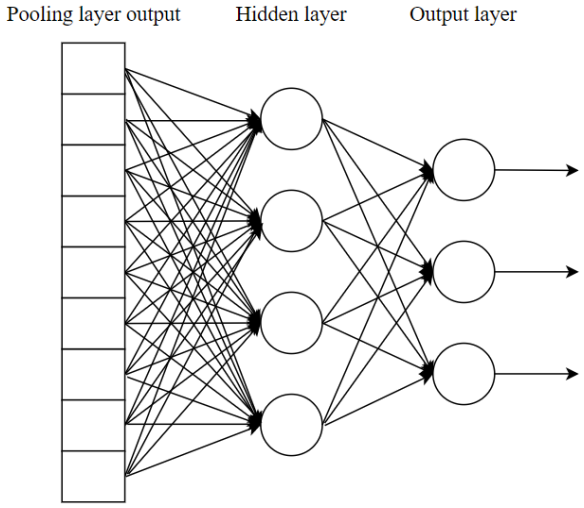

Fig. 6 Schematic diagram of fully connected layer

5. Experiment and analysis

5.1 Simulation environment

In this experiment, the mininet tool was used in the virtual machine of the ubuntu system to simulate the network environment. Mininet can create a virtual network including hosts, switches, controllers and links. Its switches support OpenFlow and are a highly flexible custom software-defined network. In the experiment, the SDN controller uses a floodlight controller written in java language. The Floodlight controller has become one of the current mainstream SDN controllers due to its open source, stability and ease of use. It can centrally and flexibly control the SDN network, providing a good expansion platform for core network and application innovation. The network structure used in this experiment is shown in Fig. 7:

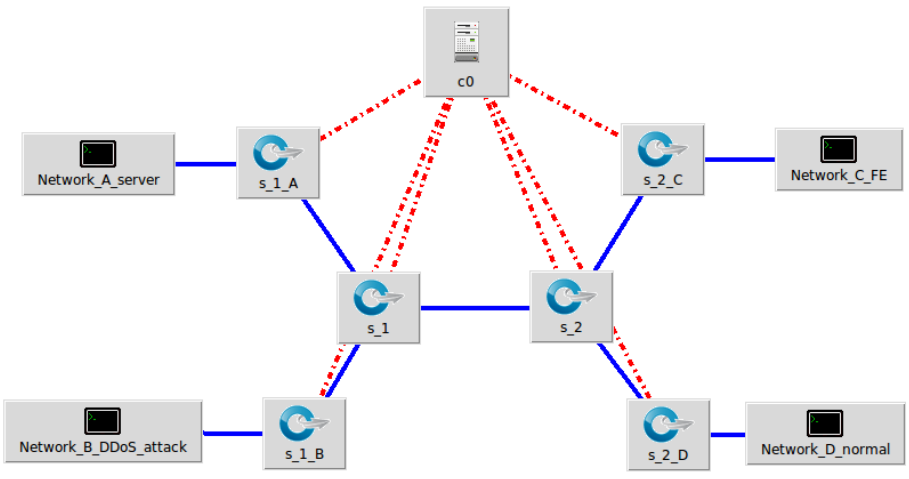

Fig. 7 Network structure in the experiment

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

During the experiment, computer group A acts as a server, computer group B injects attack traffic into the network through the hping3 command, computer group C injects Flash Event traffic into the network, and computer group D injects normal traffic. Each computer group contains several computers. The DDoS attack data sets are collected through experiments. Hping3 is a command-line oriented open source tool for generating and parsing TCP / IP protocol packet assembly / analysis, which can realize flooding denial of service attack. Therefore, hping3 can meet the experimental requirements for DDoS event simulation. After the experiment, we get 98061895 UDP attack type flow tables, 98046405 ICMP attack type flow tables and 98056793 TCP attack type packets. The data set used by Flash Event traffic is 1998 World Cup web site access Logs, this data set contains all the network requests in 80 days during the 1998 World Cup, which is a typical Flash Event data set. Because the purpose is to highlight the characteristics of Fe events, the 66 day log data with the largest daily traffic is selected as Fe data, which contains 73291868 request data, and the normal traffic uses ordinary websites for one day Log file, which contains 593757 request data. The flow table data was collected every 2 seconds to generate feature data. Finally, the data for model training are shown in Table 2.

Table 2. Data used for model training

Type of data	Data volume of training set(piece)	Data volume of test set(piece)
DdoS attack	22400	9600
Flash Event	30240	12959
Normal traffic	22566	9670

5.2 Experimental analysis

This experiment will be evaluated by accuracy, recall, precision and F1 score. The accuracy rate is used to evaluate the overall performance of the classification model. The higher the accuracy rate, the better the classification effect. F1 score, recall rate, precision rate can be observed in three aspects of how well the classification model is classified for each category. The specific calculation formula is as follows:

$$\text{Accuracy} \quad acc=\frac{TP+TN}{TP+FP+FN+TN} \tag{11}$$

$$\text{Precision} \quad P=\frac{TP}{TP+FP} \tag{12}$$

$$\text{Recall} \quad R=\frac{TP}{TP+FN} \tag{13}$$

$$F_1=\frac{2*R*P}{R+P} \tag{14}$$

In the formula, TP is the number of correct classifications classified as correct, TN is the number of correct classifications classified as wrong, FP is the number of incorrect classifications classified as correct, and FN is the number of incorrect classifications classified as wrong. At the same time, support vector machine SVM algorithm and KNN algorithm are also used as the control group of this experiment. The specific results are shown in Table 3:

Table 3. SVM algorithm, KNN algorithm and 1DCNN classification results

	Type of data	Accuracy	Precision	Recall	F1
1DCNN		0.9670	0.9672	0.9667	0.9671
SVM	DdoS attack	0.8742	0.8804	0.8726	0.8765
KNN		0.9120	0.9086	0.9108	0.8992
1DCNN		0.9542	0.9538	0.9532	0.9534
SVM	Flash Event	0.8927	0.9273	0.8043	0.8554
KNN		0.8530	0.8537	0.8518	0.8528
1DCNN		0.9641	0.9647	0.9629	0.9638
SVM	Normal traffic	0.8856	0.9193	0.8544	0.8851
KNN		0.9245	0.9278	0.9183	0.9230

ISSN: 2414-1895 DOI: 10.6919/ICJE.202106_7(6).0063

From the results of the experiment, it can be seen that compared to the KNN algorithm and the SVM algorithm, the one-dimensional convolutional neural network is better than the other two algorithms in accuracy, precision, recall rate and F1 score data, which can effectively reduce DDoS attacks. , Flash Event and Normal traffic are distinguished.

6. Conclusion

This paper proposes a DDoS attack and FE event detection method for SDN networks. The method is based on a one-dimensional convolutional neural network. By processing relevant information in the flow table, including features such as source/destination ip entropy based on φ-entropy, Effectively distinguish DDoS attacks, FE events and ordinary traffic. The advantage of this method is that it covers a wider range of network events, and has higher accuracy, precision, recall and F_1 scores than the classic KNN and SVM methods. It plays an important role in dealing with increasingly complex network environments. effect.

Due to time and knowledge constraints, this article has many areas for improvement, including finding and using updated Flash Event data sets, optimizing and finding more suitable characteristics of different traffic.

References

[1] Information on https://securelist.com/ddos-attacks-in-q1-2020/96837/

[2] Information on https://aws-shield-tlr.s3.amazonaws.com/2020-Q1_AWS_Shield_TLR.pdf

[3] Bhandari, A., Sangal, A. L., and Kumar, K. (2016) Characterizing flash events and distributed denial-of-service attacks: an empirical investigation. Security Comm. Networks, 9: 2222– 2239.

[4] M. Sachdeva, K. Kumar, G. Singh, A comprehensive approach to discriminate DDoS attacks from flash events, J. Inf. Secur. Appl. 26 (2016) 8–22.

[5] S. Behal, K. Kumar, Detection of DDoS attacks and flash events using information theory metrics-An Empirical Investigation, Comput. Commun. 103 (2017)18–28.

[6] Wentian Jiang, Yu Gu, Danni Ren, Huakang Li, Guozi Sun. DDoS attack and flash event detection based on stream characteristics in SDN[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2019,31(03):420-426.

[7] Chuanhuang Li, Yan Wu, Zhengzhe Qian, Zhengjun Sun, Weiming Wang. DDoS attack detection and defense based on deep learning hybrid model under SDN[J]. Journal of Communications, 2018, 39(07): 176-187.

[8] Xiaorui Wang, Lei Zhuang, Ying Hu, Guoqing Wang, Martin, Chenkai Jing. DDoS attack detection method based on BP neural network in SDN environment[J]. Application Research of Computers, 2018, 35(03):911-915.

[9] Sunny Behal, Krishan Kumar, Monika Sachdeva. Characterizing DDoS attacks and flash events: Review, research gaps and future directions[J]. Computer Science Review, 2017.

[10] Qingyun Zuo, Ming Chen, Guangsong Zhao, Changyou Xing, Guomin Zhang, Peicheng Jiang. SDN Technology Research Based on OpenFlow[J]. Journal of Software, 2013,24(05):1078-1097.

[11] Lingjiao Wang, Congxia Lu, Hua Guo. Research on DDoS Attack Detection Based on Support Vector Machine in SDN Environment[J]. Journal of Yunnan University (Natural Science Edition), 2021, 43(01): 52-59.