A Cooperative Detection of DDoS Attacks Based on CNN-BiLSTM in SDN

Hongwei Zhou

***Abstract*—In view of the problem that detecting DDoS attack traffic in traditional SDN depends on the controller continuously collecting traffic and running the detection model, resulting in excessive controller overhead, low detection efficiency, increased traffic forwarding delay, and easy to cause "single point of failure", a cooperative detection method of DDoS attack in SDN based on information entropy and deep learning is proposed, which divides part of the detection task into the data plane for detection based on information entropy and uses the improved CNN-BiLSTM model to detect DDoS attack traffic on control plane. The experimental results show that, compared with the SVC-RF method in recent years, the accuracy of the proposed CNN-BiLSTM model is increased by 0.74%, the detection rate is increased by 1.42%, and the false alarm rate is reduced by 1.5%. Compared with the BiLSTM model, the accuracy is increased by 0.75%, the detection rate is increased by 0.64%, and the false alarm rate is reduced by 1.14%. Compared with the RF method, the accuracy is increased by 2.34%, the detection rate is increased by 3.88%, and the false alarm rate is reduced by 4%. Compared with the traditional single point detection method which only depends on the controller, the proposed switch-controller cooperative detection method reduces the CPU occupancy of the controller by about 12% and the detection time by about 13 seconds.**

***Index Terms*—Anomaly detection, distributed denial of service attacks, deep learning, software defined network**

I. Introduction

Software defined Network is a kind of network architecture rising in recent years. Its core idea is to decouple the control plane from the data plane, separate the logical control of the network state, and abstract the controller from the underlying network equipment. The emergence of SDN greatly improves the manageability, scalability and flexibility of the network. However, with the popularity of SDN applications, the security of SDN has become one of the key research topics of SDN.

Distributed denial of Service attack, as one of the most important security threats to the Internet, is particularly dangerous in SDN because of its strong destructive power, simple implementation and lack of effective countermeasures. In recent years, attackers also began to use a variety of new devices to launch DDoS attacks on the SDN network. After the DDoS attack, the controller resources were exhausted, unable to provide normal network services for the SDN network, resulting in the paralysis of the entire SDN network.

At present, there are two main DDoS attack traffic detection methods in SDN, one is based on information entropy, and the other is based on machine learning. The detection method based on information entropy has the advantages of fast detection speed, simple and light weight, but the false alarm rate is high. The accuracy of the detection method based on machine learning is higher than that based on entropy, but it depends on the extracted traffic characteristics, and the detection speed is slower than the detection method based on information entropy. Therefore, how to design a DDoS attack detection method with high accuracy and fast detection speed has become a research hotspot.

Mousavi *et al*. proposed to calculate the information entropy of the destination IP address and compare it with the specified threshold to determine whether the network is attacked by DDoS [1]. This method is prone to false positives. On this basis, Ma et al. proposed to calculate the information entropy of the source IP address and the destination IP address of the traffic, and compare it with the specified threshold to determine whether the network is being attacked by DDoS [2]. This method increases the entropy calculation of the source IP address, but the threshold cannot dynamically adapt to the change of network size and is prone to false positives. On this basis, JunJ *et al*. added "packet speed" as a feature to detect DDoS attacks. Basicevic *et al*. [3] proposed a method of detection using generalized information entropy, but this increases the computational cost [4].

Only through the method of calculating entropy, it has the advantage of fast detection time, but it will produce a high false alarm rate, and with the change of network scale, the determined threshold should change flexibly, so its scalability is poor.

Jin *et al.* proposed to extract the 6-tuple features of the switch flow table in SDN and combine the SVM algorithm to detect the traffic, the accuracy of this method depends on the selected features [5]. Marcos *et al*. proposed a SDN defense system based on single IP traffic record analysis [6]. The system uses GRU (Gate Recurn Units) deep learning method to detect DDoS attacks, which takes a long time to detect. James et al. proposed to use RNN-LSTM to detect DDoS attack traffic, but the network model is more complex and takes a long time to detect [7]. Nisha *et al.* proposed to use a hybrid model of support vector classifier and random forest (SVC-RF) to detect traffic, first using SVC for classification, and then using random forest reasoning to classify points that cannot be inferred to belong to any class [8]. Auther *et al.* use deep neural network (DNN) to detect traffic [9]. Nisha et al. proposed to apply stackable automatic encoder multilayer perceptron (SAE-MLP) to detect DDoS attacks, they do not consider the detection time, but only consider the accuracy of

Manuscript received July 23, 2022; revised August 30, 2022; accepted October 30, 2022.

Hongwei Zhou is with School of computer Science of Guangdong University of Technology, Guangzhou, China. E-mail: 434895488@qq.com (H.W.Z.)

doi: 10.18178/ijfcc.2023.12.2.600

detection [10].

The detection accuracy of machine learning method is higher than that of entropy-based detection method, but there are still some problems, such as low detection accuracy, high detection overhead, slow detection speed and so on. As the collection, statistics and detection of traffic information need to be carried out on the SDN controller, when the scale of the network continues to expand, the controller will inevitably face huge overhead, resulting in attack detection delay [11, 12].

Aiming at the problems existing in the above research methods, this paper proposes a collaborative detection method combining information entropy and deep learning, which combines the advantages of the two detection methods. The advantages of the proposed method are as follows:

1) Different from existing methods, this method transfers part of the detection task to the switch, which can realize cross-plane detection of DDoS attack traffic. On the premise of ensuring the accuracy, this method reduces the detection overhead of the controller and improves the detection efficiency.
2) The control plane uses the improved CNN-BiLSTM model instead of the traditional machine learning method for detection, and introduces the batchnormalization mechanism to solve the problem of gradient disappearance in training deep neural network, which can learn the spatial-temporal characteristics of traffic data more comprehensively.Compared with the traditional detection method based on machine learning model, it has higher accuracy and lower false positive rate.
3) Compared with the traditional method of verification only by running datasets, a simulation platform is built in this paper to further demonstrate the effectiveness of the proposed collaborative detection method.

II. Related Work

A. CNN Model

CNN is a classification model which is widely used in the field of computer vision. One of its important features is that it can automatically detect important features without any human intervention. When CNN is applied to one-dimensional data, a variant of two-dimensional convolution is used, and the other layers remain the same. There are mainly two kinds of network layers in CNN, namely convolution layer and pooling layer. Convolution layer is usually used to extract high-dimensional features so as to reduce the number of features. Pooling layer can abstract the original features, reduce training parameters, and improve the robustness of the extracted features [13].

For the network flow data in SDN, the x_i of each stream is represented by a k-dimensional vector, where $x_i \in R^k$. Then the stream data with n characteristics can be represented as $x_{1:m} = (x_1, x_2, ..., x_n)$ as input. Conv1D uses a convolution core $\omega \in R^{hk}$ as a filter, which h represents a set of characteristics of a data stream. After you operate on the input vector, a new feature mapping is generated using the zfollowing formula:

$$c_i = f(\omega \cdot x_{i:i+h-1+b}) \tag{1}$$

where $b \in R$ represents the offset term and f represents the ReLU nonlinear function.

The filter operates on the feature set $\{x_{1:h}, x_{2:h}, ..., x_{n-h+1}\}$ of each stream data to generate a new feature mapping $c = [c_1, x_2, ..., x_{n-h+1}]$, where $c \in R^{n-h+1}$.

After the convolution layer, the pooling operation is carried out to improve the robustness of the extracted features, weaken the degree of overfitting of the model. Finally get the maximum value of all neurons in a specific area:

$$\tilde{C} = \max\{C\} \tag{2}$$

Through the above operations, the most important features are selected.

B. BiLSTM Model

LSTM is very suitable for dealing with sequence events, and network traffic data has time series characteristics. The detection of network traffic depends not only on the characteristics of current network traffic information, but also on the characteristics of past network traffic information [14, 15].

The calculation of each LSTM unit can generally be defined as follows:

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{3}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh(W_c \bullet [h_{t-1}, x_t] + b_c) \tag{5}$$

$$C_t = f_t * \tilde{C}_{t-1} + i_t * \tilde{C}_t \tag{6}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{7}$$

$$h_t = o_t * \tanh(C_t) \tag{8}$$

which f_t represents the forgetting gate, i_t represents the input gate, $\tilde{C}_t$ represents the last-minute unit state, C_t represents the unit state, o_t represents the output door, h_{t-1} represents the output of the previous unit, h_t represents the output of the current unit, W_f, W_i, W_c, W_o represents the weight of the neural network, b_f、b_i、b_c、b_o represents the offset value[17].

BiLSTM is an improved version of LSTM, which is composed of forward LSTM and backward LSTM, which can better analyze the two-way information of traffic data and perform more fine calculations. The calculation process is as follows:

$$\vec{h}_t = f(\vec{W} \bullet x_t + \vec{W} \bullet \vec{h}_{t-1} + \vec{b}) \tag{9}$$

$$\overleftarrow{h}_t = f(\overleftarrow{W} \bullet x_t + \overleftarrow{W} \bullet \overleftarrow{h}_{t-1} + \overleftarrow{b}) \tag{10}$$

$$y_t = g(U \bullet [\vec{h}_t; \overleftarrow{h}_{t-1}] + c) \quad (11)$$

$\vec{h}_t$ and $\overleftarrow{h}_{t-1}$ represent the output result of the t moment of the two LSTM layers, and $\overrightarrow{W}$ 和 $\overleftarrow{W}$ represent the hidden layer parameters of the network, x_t represent the input data, $\vec{b}$ and $\overleftarrow{b}$ represent the offset value, y_t represent the output value of the BiLSTM.

III. Method Description

The method proposed in this paper consists of two modules: initial detection module and deep detection module. The processing flow of this method is shown in Figure 1.

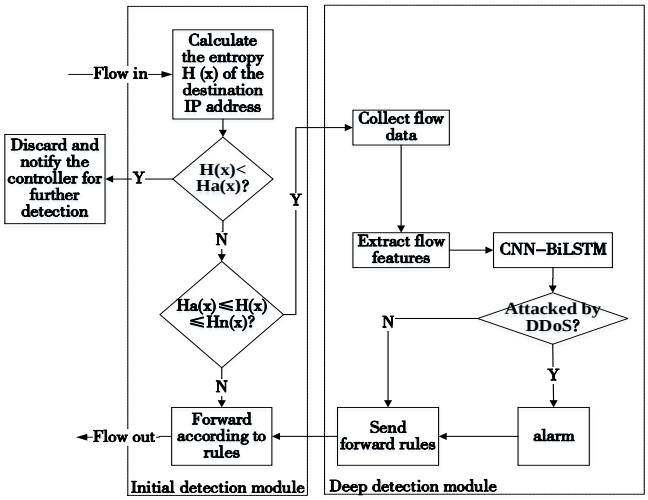

Fig. 1. The DDoS attack detection program.

The initial detection module in the switch uses the detection method based on information entropy to detect the traffic in real time. Once the abnormal traffic is found, the controller is notified immediately. When the network state is normal, the controller sends forwarding rules to the switch according to the forwarding policy to guide the forwarding of traffic, which is always in the normal working state. When the controller receives the abnormal report message sent by the switch, it starts the deep detection module and uses the CNN-BiLSTM model to further detect the abnormal traffic. If the DDoS attack traffic is detected in the network, it will start the deep detection module. The controller will alarm the administrator and issue forwarding rules to the switch to discard the data in the window.

A. Initial Detection Module Based on Information Entropy

Information entropy is a measure of the uncertainty of random variables, in which the greater the randomness of the variable, the greater the entropy value. In contrast, the higher the certainty of the information variable, the smaller the entropy [18-23]. When there is DDoS attack traffic in the network, because the DDoS attack traffic is mainly caused by the attacker sending a large number of flooding packets to a specific device by controlling the zombie device, the DDoS attack traffic often has the characteristic of a single destination IP address, which reduces the randomness of the network traffic. Therefore, when there is DDoS attack traffic in the network, the entropy of all destination IP addresses in the network is lower than that when the network state is normal.

Suppose that the collection of destination IP addresses of n different packets $S = \{S_1, S_2, ..., S_i, ..., S_n\}$ received by the switch, S_i represents the sum of all packets with the destination IP address. $P_i = \dfrac{S_i}{\sum_{i=1}^{n} S_i}$ represents the probability of occurrence of the IP address for the i[th] destination. The formula of Shannon entropy is defined as:

$$H(x) = -\sum_{i=1}^{n} p_i \log_2 p_i \quad (12)$$

Eq. (12) shows that when the value of the sample is more concentrated, the entropy value is smaller; when the value of the sample is more dispersed, the entropy value is larger. In order to facilitate comparison, after the entropy value is calculated, it is necessary to normalize the information entropy, as shown in the Eq. (13):

$$H'(x) = \frac{H(x)}{\log_2 n} \quad (13)$$

In the experiment, the received data packet is observed based on the window value, and the n in the Eq. (12) is selected to calculate the entropy of the destination IP address of the received packet.

Suppose that in the event of a DDoS attack, the threshold for the entropy of the destination IP address of the received packet is $H_a(x)$. The threshold of the entropy of the destination IP address of the received packet is $H_n(x)$ when the network condition is normal. The entropy of the destination IP address of a packet received by a window on the switch is calculated as $H(x)$. It is judged as DDoS attack traffic when $H(x) < H_a(x)$,then the data in the window is discarded and an abnormal report message is sent to the controller, and the deep detection module is started; if $H_a(x) < H(x) < H_n(x)$, it is determined that there is abnormal traffic but not discarded, an abnormal report message is sent to the controller to start the deep detection module; if $H(x) > H_n(x)$, it is determined to be normal traffic, and normal forwarding rules are requested from the controller. The pseudo code of the initial detection module is shown in algorithm 1.

Algorithm 1
Input: FlowData
Output: Action
1: Initialize entropy threshold: *Hn*(*X*), Ha(*X*)
2: start:
3: for *i*=1: *n* do
4: Calculate the entropy *H* (*X*) of n IP addresses
5: end for
6: if *Ha*(*X*) > *H*(*X*) then
7: Send an exception report message to the controller
8: Discard the data in the window
9: elseif *Ha* (*X*) < *H*(*X*) < *Hn*(*X*) then

10: Send an abnormal report message to the controller
11: else
12: goto start //Enter the calculation of next window
13: end if

The process of implementing the data plane in P4 language is as follows:

1) Define related data structures such as IP, TCP, UDP, ethernet frame packet header, metadata generated during P4 program execution, and data structure of ddosd protocol. Among them, the metadata should contain fields such as the number of different IP addresses, the information entropy of the destination IP address, and the initial detection result. And the ddosd data should include the number of data packets, the initial detection result, the information entropy of the destination IP, the frame type and other fields. The initial detection result field is used to inform the controller to start the deep detection module.
2) Define the module to parse the data, and set the parsed protocol type to 0x0800 and 0x6605, which represent IP protocol data and custom ddosd protocol data respectively.
3) Define the ingress module, in which the detection method based on information entropy is implemented in this module, the count-min sketch method is used to realize the statistics of the occurrence frequency of data packets, and the hash values of different data packets are stored through a two-dimensional address space, and the hash values of many kinds of data packets may be the same, so the actual value may be too large. Set up several different hash functions and take the minimum hash value to reduce the error. The new formula for calculating information entropy is:

$$H(X)=\log_2(n)-\frac{1}{n}\sum_{x=1}^{N} f_x \log_2(f_x) \tag{14}$$

In the Eq. (14), $S=\sum_{x=0}^{N} f_x \log_2(f_x)$ represents the entropy norm, n is the total number of packets, N represents the number of different destination IP addresses, and the number of occurrences of each different destination IP address is f_x.

After completing the above definition, it is necessary to determine the window size, and the accuracy under different windows is tested many times with the sensitivity coefficient. Finally, 128 is selected as the window value. If the entropy value of the destination IP address in a window is lower than the set threshold range, the traffic data marked in the window is abnormal traffic and notifies the controller.

4) Define the egress module and complete the encapsulation of the custom ddosd protocol data packet according to the calculation results.

B. Deep Detection Module Based on CNN-BiLSTM

1) Feature extraction

How to select the flow characteristics will have a great impact on the detection model, which may increase the complexity of the model, bring greater overhead to the model, and affect the efficiency of detection [24]. After receiving the exception report message from the switch, the controller sends a request message to the switch through the OpenFlow protocol to count the flow table information, such as the number of packets, the number of data bytes, the port number, the source IP address and so on. As the input of the CNN-BiLSTM model in the deep detection module, some fields in the flow table items are selected as the features extracted directly. In addition, some features that need to be calculated are added, with a total of 19 features, as shown in Table I:

TABLE I: THE ARRANGEMENT OF CHANNELS

Feature Name	Description
Src_ip	Source ip address
Dst_ip	Destination ip address
PktCount	The number of packets contained in all flows on the switch
ByteCount	The number of bytes contained in all flows on the switch
Duration_sec	Duration of the flow
Switch-id	Switch id
TX_Bytes	Number of bytes sent from the switch port
RX_Bytes	Number of bytes received from switch port
Port Number	Port number
Src_mac	Source mac address
Dst_mac	Destination mac address
Packet Rate	Average rate of packets
Flows	Total number of flows on the switch
Protocol	Protocol of flow
Tx_kbps	Data transmission rate
Rx_kbps	Data receiving rate
Tot_kbps	Total number of Tx_kbps and Rx_kbps
Pktperflow	Number of packets per flow
Byteperflow	Number of bytes per flow

2) Batch normalization

In CNN-BiLSTM, the data distribution in the deep neural network may change after the traffic data is extracted by BiLSTM. In order to solve the problem of inconsistent data distribution in the deep neural network, the Batch Normalization mechanism is introduced to speed up the training speed of the neural network. It normalizes the data input from the previous layer after the activation function is non-linearly transformed. Thus, the trainability of the neural network can be ensured, and the neural network can always maintain the consistency of the distribution of the input data, thus reducing the great transformation of the node distribution within the network. The use of Batch Normalization mechanism can not only accelerate the convergence speed of the network, but also maintain the representation ability of the neural network.

The calculation method for each layer of Batch Normalization is as follows:

$$B=\{X_{1...m}\} \tag{15}$$

$$y_i = BN_{\lambda,\beta}(x_i) \tag{16}$$

$$\mu_\beta = \frac{1}{m}\sum_{i=1}^{m} x_i \tag{17}$$

$$\sigma_\beta^2 = \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_\beta)^2 \tag{18}$$

$$x_i^{'} = \frac{x_i - \mu_\beta}{\sqrt{\sigma_\beta^2 + \varepsilon}} \tag{19}$$

$$y_i = \gamma * x_i^{'} + \beta \tag{20}$$

where B indicates that there are m activation values in a batch, $x_i^{'}$ represents the normalized value, and y_i represents the transformed value BN.

3) CNN-BiLSTM module

Through the feature extraction module, the 19 features shown in Table I are extracted, and then the data is preprocessed as the input of the CNN-BiLSTM model to further determine whether there is DDoS attack traffic in the network. If there is DDoS attack traffic, update the flow table immediately and guide the switch to discard the data in the detection window. Otherwise, the forwarding rules will be sent normally.

The training process of the CNN-BiLSTM model used is shown in Fig. 2.

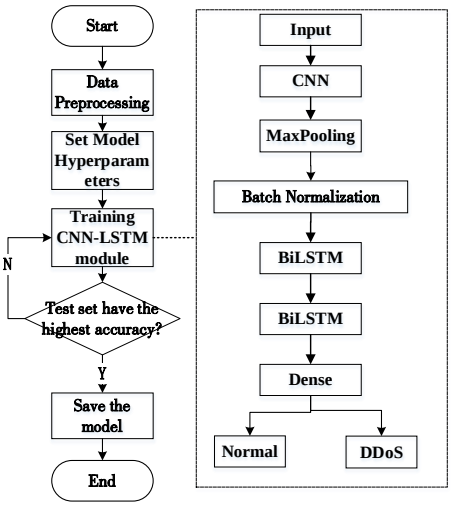

Fig. 2. The CNN-BiLSTM model training process.

The feature vector is first input into the CNN layer to extract spatial hierarchical features and control the ability of model fitting, and then sampled through the MaxPooling layer, so as to reduce the size of the model, improve the computing speed and improve the robustness of the extracted features. Next is the Batch Normalization layer, which is responsible for normalizing each input batch of the model to avoid the problem of gradient disappearance in the training process. After processing, input to the BiLSTM layer to capture the forward time feature series information and reverse time feature series information of the network traffic data, and discover the structural characteristics of the abnormal traffic data. Finally, through the Dense layer, the purpose is to non-linearly transform the previously extracted features in order to obtain the correlation between these features, and finally map to the output space to output classification result.

IV. Experiment and Result Analysis

A. Experimental Environment

The hardware parameters used in the experiment are as follows: CPU is AMD Ryzen 7 4800H@4.20 GHz 8 core, memory is 16GB DDR4 3200MHz RAM, operating system is Ubuntu 16.04LTS.

The initial detection module based on information entropy is developed in P4 language, P4c tool is used as compiler, switch is BMv2 switch, v1 model is used as the architecture foundation of data plane, and P4RuntimeAPI is used as the interface between control plane and data plane. The deep detection module based on CNN-BiLSTM model is developed using Kreas deep learning framework. Use mininet to build a network topology to simulate a real network environment, as shown in Fig. 3.

DDoS attack traffic is generated using the tool TFN, including mixed traffic of SYN flood and ICMP flood. H1 is selected as the victim host and H7 as the attack source to generate DDoS attack traffic. In order to be closer to the real network environment, D-ITG tool is used to generate background traffic to simulate the normal communication of the real network.

The written P4 initial detection program .json file is generated by the compilerand, and put into switch S1 for automatic configuration. The RYU controller uses the trained CNN-BiLSTM model for further detection.

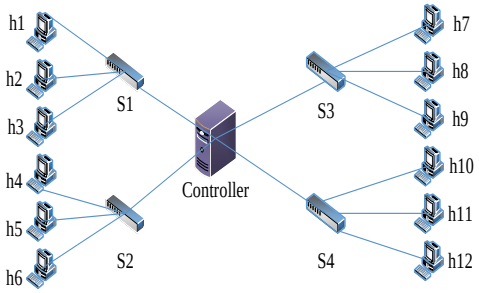

Fig. 3. The SDN environment simulations in the mininet.

B. Analysis of the Experimental Results of the Initial Detection Module Based on Information Entropy

According to the repeated experiments and historical statistics, when the network state is normal, the entropy threshold of the destination IP address is set to 0.75. When there is DDoS attack traffic in the network, the entropy threshold of the destination IP address is set to 0.55.

TABLE II. Judging Rule

Entropy Range	Judgment Result	Action of Switch
$0.75 \leqslant H(x)$	Normal	Normal forwarding
$0.55 < H(x) < 0.75$	Abnormal	Report controller
$H(x) \leqslant 0.55$	DDoS	Drop the traffic and report controller

When using TFN tool to generate DDoS attack traffic injected into the network, in order to better show the trend of the curve, the entropy of data packets in the window is

recorded every 1 second, and the interval of no record entropy is regarded as the same as the previous record, and the decision rules are shown in Table II.

As shown in Fig. 4, the curve is above 0.75 until the DDoS attack traffic is generated. At the 11th second, the network begins to simulate the generation of DDoS attack traffic, and the curve is between 0.55 and 0.75, and gradually falls below 0.55, the switch begins to drop the traffic, and it can be observed that the entropy of the destination IP address gradually picks up until the 30th second, the entropy of the destination IP address returns to more than 0.75.

It can be concluded from Fig. 4 that when a DDoS attack occurs in the SDN network, the initial detection module based on the destination IP information entropy on the switch can obviously detect the change of entropy and initially identify the DDoS attack traffic in the SDN network.

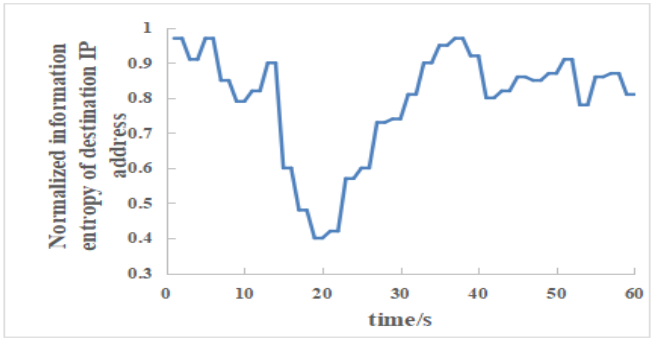

Fig. 4 Information entropy change of the destination IP address.

Fig. 5 shows the comparison of the number of Packet-in packets received by the controller when there is an initial detection module and no initial detection module on the switch. The simulation of generating DDoS attack traffic begins in 11 seconds. It can be found that when there is an initial detection module, the peak value of the curve and the rate of reaching the peak are lower than without initial detection module. At the same time, the number of Packet-in packets received by the controller returns to the normal value by about 2 seconds ahead of schedule. The main reason is that once the destination IP entropy of the traffic detected in the initial detection module is less than 0.55, it will directly determine that the received data is DDoS attack traffic, thus directly discarding the traffic, which plays the role of early mitigation. When there is no initial detection module, it is necessary to wait for the result of CNN-BiLSTM model detection in the controller to be discarded. Because CNN-BiLSTM model detection takes a long time, there is a certain delay. Therefore, the efficiency of emergency response with initial detection module is higher than that without initial detection module.

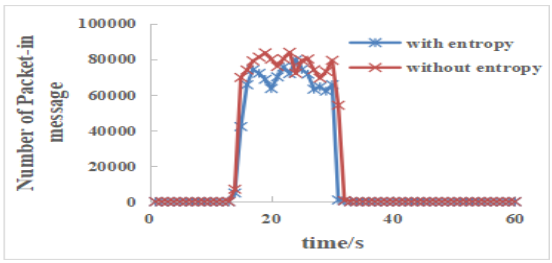

Fig. 5. Number of Packet-In packets changes in the controller.

Looking at the comparison of figure 6, the overall CPU occupancy rate of the controller with the initial detection module on the switch is lower than that without the initial detection module.

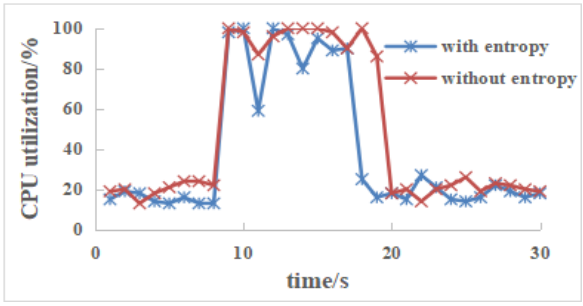

Fig. 6. The CPU occupancy change of the controller.

Analyzing the reason, when the network state is normal (0s-9s, 20s-30s), only the initial detection module is used to detect when the switch has an initial detection module. The controller will start the deep deteciton module only after receiving the abnormal report message from the switch, so the CPU utilization rate is low. When there is no initial detection module on the switch, we can only rely on the controller to use CNN-BiLSTM model to detect network traffic in real time, and the detection cost of CNN-BiLSTM model is high, so the utilization rate of CPU is higher. When there is DDoS attack traffic in the network (9s-20s), due to the high rate of DDoS attack traffic, both schemes require a certain reaction time, and the peak value is 100%. However, the CPU utilization rate of the controller with the initial detection module on the switch returns to the normal range about 2s earlier than that without the initial detection module. The main reason is that when there is an initial detection module, the switch will determine the existence of DDoS attack traffic in the network when the destination IP entropy of the traffic is lower than 0.55, and directly discard the subsequent received traffic. Therefore, the number of Packet-In messages sent to the controller to process is reduced, so that the CPU utilization rate of the controller can be restored to the normal range more quickly. When there is no initial detection module on the switch, because we have to wait for the detection result of the CNN-BiLSTM model before discarding traffic, the switch keep sending Packet-in messages to the controller, which makes the controller deal with Packet-in messages consistently, so the CPU utilization rate of the controller will be delayed back to the normal range.

In order to compare the cost of the proposed switch and controller cooperative detection method and the traditional controller centralized detection method . By changing the size of the network, only the normal network traffic is simulated for 30 minutes, and the CPU utilization rate of the controller is recorded every 5 minutes. Finally, the average CPU utilization rate of the controller is calculated. The experimental results are shown in Table III.

TABLE III. DIFFERENT NETWORK SIZES HAVE AN INFLUENCE ON THE CONTROLLER OVERHEAD OF THE TWO METHODS

Total number of terminals and switches	Average CPU utilization of the controller /%	
	Cooperative detection	Centralized detection
16	25	37
32	36	50
40	42	60

It can be concluded from Table III that the CPU utilization of the controller of the cooperative detection method is lower than that of the centralized detection method, especially when

the network scale increases, the cooperative detection method reduces the CPU utilization of the controller more obviously.

From the above experimental results, it can be seen that the cooperative detection method requires less overhead and higher efficiency than the traditional centralized detection method. Because the threshold of information entropy is not constant, it will be affected by the network scale and the dynamic change of network traffic， and the information entropy of normal traffic IP may be lower than the threshold of IP information entropy of abnormal traffic. Therefore, we can not only rely on the detection method based on information entropy to detect DDoS attack traffic, but as a way to reduce controller overhead.

C. Analysis of Experimental Results of Deep Detection Module Based on CNN-BiLSTM Model

1) Dataset

In order to make the experimental environment closer to the real environment, the model is trained and tested using the public dataset sdn_dataset in [16]. The dataset contains a total of 104345 pieces of flow data, including 63561 pieces of normal flow data and 40784 pieces of DDoS attack flow data. The number of normal and attack samples is distributed as shown in Table IV.

TABLE IV. DIVISION OF THE TRAINING SET AND THE DATASET

Sample	Training Set	Test Set
Normal sample	50848	12713
Attack sample	32627	8157

2) Data preprocessing

There are 23 kinds of traffic characteristics in the dataset, 19 kinds of features are extracted as shown in Table 1, and one-hot codes are performed on Source_IP, Destination_IP, Protocol and other features that do not have numerical values. Other features that are too large or too small need to be normalized, in which the high-value feature and the low-value feature are scaled to the value between (0,1) after preprocessing, due to the addition of pseudo-variable coding for non-numerical features in the dataset, the dataset includes a total of 65 columns, which are converted into 65-dimensional feature vectors and input to the CNN-BiLSTM model for training and testing.

3) Training model

Through grid search and continuous tuning of hyperparameters, it is determined that the batch size is 64, the learning rate is 0.01 and the number of training iterations is 50. The CNN-BiLSTM model mainly consists of two hidden BiLSTM layers, and the number of nodes in each BiLSTM hidden layer is 192. After 19-element feature extraction and preprocessing, the feature extraction module forms a 65-dimensional feature vector and inputs it into the CNN-BiLSTM model. The classifier of the model output layer selects the sigmoid activation function, the optimization function uses Adam, and the loss objective function chooses the binary cross-entropy function. In order to prevent over-fitting of the model before the output of the Dense layer, the dropout operation is added to improve the generalization ability of the model.

The different number of nodes in the BiLSTM layer will affect the effect of the model. Using the same dataset, setting different nodes in the BiLSTM layer to compare their accuracy. After 50 training rounds, the same test set is used for testing, and the results are shown in Table V.

TABLE V. ACCURACY OF DIFFERENT NUMBERS OF NODES IN THE BILSTM LAYER

Nodes of BiLSTM Layer	ACC/%
64	97.23
128	99.19
192	99.54
256	98.62

As can be seen from the results in Table 5, when the nodes of BiLSTM layer is 192, the accuracy reaches the best. Excessive number of nodes will reduce the accuracy, so 192 is chosen as the node number of BiLSTM layer.

The number of convolution kernels will affect the extraction of data features by CNN layer. In order to study the influence of the number of convolution kernels on detection accuracy, the number of convolution kernels is set as 1 to 6 and verified successively.The experimental results are shown in Fig. 7.

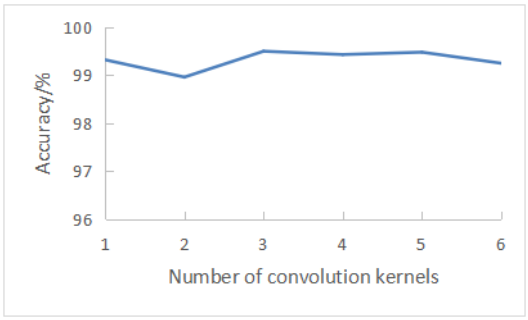

Fig. 7. Effect of the number of convolution kernels on the accuracy.

As can be seen from Fig. 7, when the number of convolution kernels is 3, the accuracy is the highest, but when the number of convolution kernels increases, the accuracy decreases slightly. Therefore, when the number of convolution kernels is 3, the experimental effect is the best.

The training results of CNN-BiLSTM model are shown in Figure 8 and Figure 9.It can be seen from Figure 8 that with the increase of training rounds, the accuracy of the model in the training set and the test set gradually increases, and the accuracy of model detection reaches the highest at the 50th training round.As can be seen from FIG. 9, with the increase of training rounds, the value of loss function also decreases gradually and reaches the minimum at the 50th round.The curve fluctuation of the two graphs is not large and gradually becomes stable, so they have a certain generalization ability.

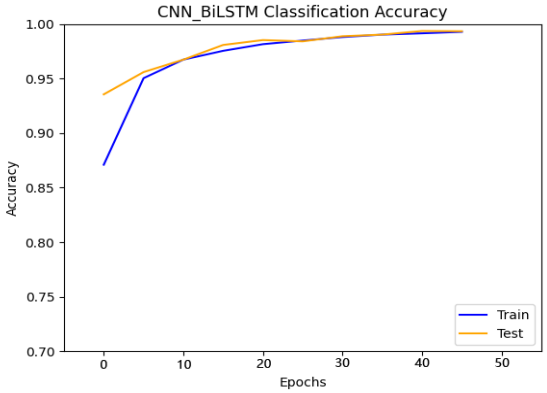

Fig. 8. Training accuracy of the sdn_dataset dataset.

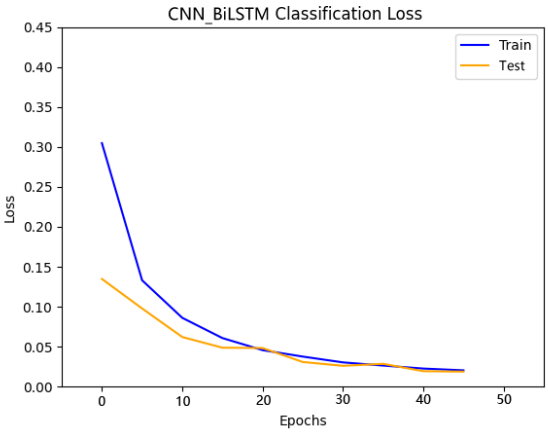

Fig. 9. Changes in the loss function.

4) Result analysis

Detection Rate (DR), Accuracy (ACC) and False Alarm Rate (FAR) were used as the evaluation criteria of the model.

$$DR = \frac{TP}{TP + FN} \tag{21}$$

$$DR = \frac{TP}{TP + FN} \tag{22}$$

$$ACC = \frac{TP + TN}{TP + TN + TP + FN} \tag{23}$$

Among them, TP represents the number of correctly classified attack samples, FP represents the number of incorrectly classified attack samples, TN represents the number of correctly classified normal samples, and FN represents the number of incorrectly classified normal samples [25].

For the same dataset sdn_dataset in [16], CNN-BiLSTM model used in this paper was compared with traditional LSTM, BiLSTM, SVC-RF method used in literature [8] and RF method used in literature [9], and the results were shown in Table VI.

TABLE VI: EVALUATION METRICS OF DIFFERENT MACHINE LEARNING MODELS AND CNN-BILSTM MODELS

Method	ACC/%	DR/%	FAR/%
LSTM	96.20	95.10	4.43
BiLSTM	98.79	98.69	1.64
RF	97.20	95.45	4.50
SVC-RF	98.80	97.91	2.00
CNN-BiLSTM	99.54	99.33	0.50

As can be seen from Table VI, compared with the SVC-RF method in the best performance literature [8], the accuracy of the CNN-BiLSTM model proposed in this paper is increased by 0.74%, the detection rate is increased by 1.42%, and the false alarm rate is reduced by 1.5%. Compared with the BiLSTM model, the accuracy is increased by 0.75%, the detection rate is increased by 0.64%, and the false alarm rate is reduced by 1.14%. Compared with the RF method in literature [9], the accuracy is increased by 2.34%, the detection rate is increased by 3.88%, and the false alarm rate is reduced by 4%.

According to the analysis of the reason, BiLSTM is composed of forward LSTM and backward LSTM, which can better analyze the two-way information of traffic data, so the accuracy is higher than LSTM. However, both LSTM and BiLSTM can only learn the time sequence characteristics of network data traffic, and network data traffic not only has the characteristics of time series, but also has the characteristics of high-dimensional space. Therefore, the CNN-BiLSTM method can learn the characteristics of network data traffic more comprehensively. For the RF method, when the data has many attributes with different values, the credibility of the attribute weights generated by the RF algorithm will decrease, while the network traffic data in SDN will have such attributes, such as the number of bytes sent by the port, the transmission rate of the port, the number of bytes per stream and other characteristics will change with the change of the network state, so the accuracy will be low. For SVC-RF method, we first use SVC to classify network traffic data, and then use RF to further classify some misclassified data near the classification hyperplane. Because the amount of data is reduced, the influence of the credibility of attribute weights generated by RF algorithm is reduced, and the misclassified data can be classified correctly again.Therefore, the effect of SVC-RF method is better than that of RF method. Compared with SVC-RF method and CNN-BiLSTM method, SVC-RF method relies more on manual feature extraction, while CNN-BiLSTM method can automatically learn the high-dimensional spatial features and bi-directional time series features of network traffic data. In addition to the features extracted manually, it can further learn the potential characteristics of network data traffic, and the batch normalization mechanism is introduced to enhance the representation ability of the model. Therefore, the performance of the proposed CNN-BiLSTM method is better than that of SVC-RF method.

To sum up, the performance of the proposed CNN-BiLSTM method is better than that of all the methods mentioned above.

In order to verify the improvement of the proposed method on the controller overhead again, a control experiment was carried out using the same traffic data. First of all, only the deep detection module based on CNN-BiLSTM model is deployed to detect the traffic. Then, on the basis of the first experiment, the initial detection module based on information entropy is deployed on the switch to detect the traffic. The experimental results are shown in Table VII.

TABLE VII: WITH INITIAL DETECTION MODULE AND WITHOUT INITIAL DETECTION MODULE OVERHEAD SITUATION

Method	Average CPU Occupancy /%	Detection Time/s
CNN-BiLSTM	37	23.74
Entropy+CNN-BiLSTM	25	10.26

From the CPU occupancy of the controller, it can be seen that after the initial detection module is deployed on the switch, the overall average CPU occupancy is reduced by about 12%. From the detection time, it can be seen that after the deployment of the initial detection module, the detection time is shortened by about 13.48 seconds. On the whole, the addition of the initial detection module can reduce the

overhead of the controller and reduce the detection time.

To sum up, the cooperative detection method based on information entropy and CNN-BiLSTM model proposed in this paper is better than the comparative research method in accuracy, detection rate and false alarm rate, and compared with the traditional controller single point detection method, this method shortens the detection time and reduces the cost of the controller.

V. Conclusion

In this paper, a cooperative detection method of DDoS attacks based on information entropy and CNN-BiLSTM model in SDN environment is proposed. In this method, part of the detection task is split into the data plane, and the improved CNN-BiLSTM model is used for further detection. According to the characteristics of the destination IP address set of the DDoS attack traffic, the DDoS attack traffic is initially detected by calculating the information entropy of the destination IP address and deployed on the switch. The deep detection module detects the traffic using the method based on the CNN-BiLSTM model, and inputs the 19 features of the traffic into the CNN-BiLSTM model for detection, which is deployed on the controller. When the initial detection module finds that there is DDoS attack traffic in the network after entropy calculation, it sends an exception report message to the controller, and then starts the deep detection module for further detection. The controller does not need to use the CNN-BiLSTM model for real-time detection, thus reducing the overhead of controller detection. The experimental results show that the detection method proposed in this paper shortens the overall detection time and reduces the CPU occupancy rate of the controller, and is better than the comparative research method in accuracy, detection rate and false alarm rate. The future direction will be devoted to studying the information entropy threshold in the initial detection module, so that it can be dynamically adjusted according to the state of the network.

Conflict of Interest

The author declares no conflict of interest.

Funding

This paper was supported by Guangzhou Key Field Research and Development Project (202007010004) in China.

References

[1] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," presented at the 2015 International Conference on Computing, Networking and Communications (ICNC) IEEE, March 30, 2015.

[2] Y. H. Chen and X. L. Ma, "Chaos analysis based on Markov chains in DDoS detection," presented at the International Conference on Computer Science and Systems Engineering (CSSE), 2014.

[3] J. H. Jun, D. Lee, C.W. Ahn, and S. H. Kim, "DDoS attack detection using flow entropy and packet sampling on huge networks," presented at the Thirteenth International Conference on Networks (ICN), 2014.

[4] B. Ilija, S. Ocovaj, and M. Popovic, "Use of tsallis entropy in detection of SYN flood dos attacks," *Security and Communication Networks*, vol. 8, no. 18, pp. 3634-3640, 2016.

[5] J. Ye, X. Cheng, Z. Jian, L. Feng, and S. Ling, "A DDoS attack detection method based on SVM in software defined network," *Security and Communication Networks*, vol. 2018, pp. 1-8, 2018.

[6] M. Assis, L. F. Carvalho, J. Lloret, and M. L. Proena, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, p. 102942, 2020.

[7] A. B. Opare, "An Investigation into the application of deep learning in the detection and mitigation of DDOS attack on SDN controllers," *Technologies*, vol. 9, 2021.

[8] N. Ahuja, G. Singal, D. Mukhopadhyay, and N. Kumar, "Automated DDOS attack detection in software defined networking," *Journal of Network and Computer Applications*, vol. 187, no. 6, pp. 103-108, 2021.

[9] V. Y. Kulkarni and P. K. Sinha, "Pruning of Random Forest classifiers: A survey and future directions," presented at the International Conference Data Science and Engineering (ICDSE), 2012.

[10] D. S. M. A. Jat and A. M. Gamundani, "Deep neural network (DNN) solution for real-time detection of distributed denial of service (DDoS) attacks in software defined networks (SDNs)," *SN Computer Science*, vol. 2, no. 2, 2021.

[11] R. D. Corin *et al*., "LUCID: A practical, lightweight deep learning solution for DDoS attack detection," *IEEE Transactions on Network and Service Management*, vol. 17, no. 2, pp. 876-889, 2020.

[12] X. L. Lu, P. Liu, and J. Lin, "Network traffic anomaly detection based on information gain and deep learning," presented at the 3rd International Conference, 2019.

[13] A. Nisha, G. Singal, and D. Mukhopadhyay, "DLSDN: Deep learning for DDOS attack detection in software defined networking," presented at the Confluence, January 28, 2021.

[14] T. Liu and S. Yin, "Detection and identification of phased DDoS attacks based on cross entropy in SDN environment," *Computer Applications and Software*, vol. 38, no. 2, pp. 328-333, 2021.

[15] Ö. Tonkal, H. Polat, E. Başaran, Z. Cömert, and R. Kocaoğlu, "Machine learning approach equipped with neighbourhood component analysis for DDoS attack detection in software-defined networking," *Multidisciplinary Digital Publishing Institute*, vol. 11, 2021.

[16] A. Nisha, S. Gaurav, and M. Debajyoti, "DDOS attack SDN dataset," Mendeley Data, 2020.

[17] M. A. Khan, M. R. Karim, and Y. Kim, "A scalable and hybrid intrusion detection system based on the convolutional-LSTM network," *Symmetry*, vol. 11, no. 4, 2019.

[18] L. Zhang and J. S. Wang, "DDoS attack detection modelbased on information entropy and DNN in SDN," *Computer Research and Development*, vol. 56, no. 5, pp. 909-918, 2019.

[19] J. Zhu, Z. D. Wu, L. B. Ding, "Attack detection based on DBN in SDN environment," *Computer Engineering*, vol. 46, no. 4, pp. 157-161, 2020.

[20] D. Ding, M. Savi, and D. Siracusa, "Tracking normalized network traffic entropy to detect DDoS attacks in P4," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no 6, pp. 4019-4031, 2021.

[21] A. Ba *et al*., "A survey on DoS/DDoS mitigation techniques in SDNs: Classification, comparison, solutions, testing tools and datasets," *Computers and Electrical Engineering*, vol. 99, 2022.

[22] A. L. Ying *et al.*, "Software-defined DDoS detection with information entropy analysis and optimized deep learning," *Future Generation Computer Systems*, vol. 129, 2022.

[23] A. Aa *et al*., "A low-rate DDoS detection and mitigation for SDN using Renyi entropy with packet drop," *Journal of Information Security and Applications*, vol. 68, 2022.

[24] N. Beny and R. N. Murthy, "Deep learning-based slow DDoS attack detection in SDN-based networks," presented at the IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) IEEE, 2020.

[25] F. R. Fadaei, E. Orhan, and A. Emin, "A novel approach for distributed denial of service defense using continuous wavelet transform and convolutional neural network for software-defined network," *Computers and Security*, vol. 112, no. 3, 2022.

[26] N. Aslam, S. Srivastava, and M. M. Gore, "ONOS flood defender: An intelligent approach to mitigate DDoS attack in SDN," *Transactions on Emerging Telecommunications Technologies*, vol. 9, 2022.

[27] A. Mvod *et al*., "Near real-time security system applied to SDN environments in IoT networks using convolutional neural network," *Computers and Electrical Engineering*, vol. 86, no. 3, p. 106738, 2020.

[28] W. Zhang *et al*., "Machine learning based low-rate DDoS attack detection for SDN enabled IoT networks," *International Journal of Sensor Networks*, vol. 34, no. 1, pp. 56, 2020.

[29] K. E. Rajakumari, M. S. Kalyan, and M. V. Bhaskar, "Forward forecast of stock price using LSTM machine learning algorithm," *International Journal of Computer Theory and Engineering*, vol. 12, no. 3, pp. 74-79, 2020.

[30] M. P. Novaes *et al*., "Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments," *Future Generation Computer Systems*, vol. 125, no. 3, pp. 156-167 2021.