Mathematical Biosciences and Engineering

MBE, 21(3): 4187–4209.
DOI: 10.3934/mbe.2024185
Received: 30 October 2023
Revised: 01 February 2024
Accepted: 02 February 2024
Published: 26 February 2024

http://www.aimspress.com/journal/MBE

Research article

MFFLR-DDoS: An encrypted LR-DDoS attack detection method based on multi-granularity feature fusions in SDN

Jin Wang[1,*], Liping Wang[1,*] and Ruiqing Wang[2]

[1] College of Computer Science & Technology, Zhejiang University of Technology, Hangzhou 310023, China
[2] School of Mathematics, Zhengzhou University of Aeronautics, Zhengzhou 450046, China

* **Correspondence:** Email: 1111912010@zjut.edu.cn, wlp@zjut.edu.cn.

Abstract: Low rate distributed denial of service attack (LR-DDoS) is a special type of distributed denial of service (DDoS) attack, which uses the vulnerability of HTTP protocol to send HTTP requests to applications or servers at a slow speed, resulting in long-term occupation of server threads and affecting the normal access of legitimate users. Since LR-DDoS attacks do not need to send flooding or a large number of HTTP requests, it is difficult for traditional intrusion detection methods to detect such attacks, especially when HTTP traffic is encrypted. To overcome the above problems, we proposed an encrypted LR-DDoS attack detection and mitigation method based on the multi-granularity feature fusion (MFFLR-DDoS) for software defined networking (SDN). This method analyzes the encrypted session flow from the time sequence of packets and the spatiality of session flow and uses different deep learning methods to extract features, to obtain more effective features for abnormal traffic detection. In addition, we used the advantages of SDN architecture to perform real-time defense against LR-DDoS attacks by the way of SDN controller issuing flow rules. The experimental results showed that the MFFLR-DDoS model had a higher detection rate than advanced methods, and could mitigate LR-DDoS attack traffic online and in real-time.

Keywords: LR-DDoS attack; malicious encrypted traffic; anomaly detection; SDN; feature fusion; deep learning

1. Introduction

With the wide application of Internet technology, the environment of network equipment is becoming more and more complex. It is very difficult to manage the huge scale of the network, and many security problems have become increasingly prominent. SDN [1] is a novel network architecture that separates the control plane from the forwarding plane and is directly programmable. It abstracts traditional control plane functions from various network nodes and reconstructs the original distributed control network architecture into a centralized control network architecture. The novel and unique architecture of SDN shields the hardware differences of the underlying devices, simplifies the configuration and management of network devices, reduces the operation cost of the network, and realizes the flexible deployment of network devices. However, SDN also suffers from various attacks by hackers, and the most threatening attack is the DDoS attack.

DDoS attack is a type of attack against network services, in which attackers send a large number of request flows or establish a large number of connections to hosts or servers through a distributed zombie network to exhaust the processing power, bandwidth resources, or service capacity of the target host or server, ultimately making it unable to receive or respond to normal service requests from legitimate users. Due to the characteristics of wide attack range, simple operation, strong concealment, and difficult mitigation, DDoS attacks can cause significant harm to the network. DDoS attacks can be roughly divided into three categories: Volumetric attacks, protocol attacks, and application layer attacks [2]. Among them, application layer attacks are the most complex and severe type of attack. In this article, we discuss a type of SDN application layer DDoS attack called LR-DDoS. The LR-DDoS attack achieves resource occupancy of web servers by constructing a large number of legitimate slow requests at the application layer. Initially, the attacker utilizes the normal exchange mechanism of the HTTP protocol to establish a connection with the web server and sets a relatively large Content Length. Then, the attacker sends packets at a very low speed, such as sending a byte in 1–10 seconds and maintaining the connection continuously. If attackers establish multiple such connections through a zombie network, the available connections on the web server will gradually be occupied by the attacker, making it unable to provide services to normal users. There are various types of LR-DDoS attacks, commonly including Slowloris, Slow Post, and Slow Read [2]. Because executing LR-DDoS attacks does not require high-performance devices or consume a large amount of network bandwidth resources, that means simple network devices can achieve the effect of LR-DDoS attacks. Therefore, the low cost, slow speed, and strong concealment of LR-DDoS attacks pose new challenges to network security. It mainly includes the following three aspects: First, compared to traditional DDoS attacks, LR-DDoS attacks have similar traffic patterns to legitimate traffic patterns and even require less traffic to achieve the attack target. Therefore, traditional detection methods are not suitable for LR-DDoS attacks. Second, with the increasing emphasis on data security and privacy, the detection of encrypted malicious traffic is also receiving increasing attention. However, current research on LR-DDoS attacks is mainly used for nonencrypted attack detection, and there is little research on encryption-based LR-DDoS attack detection technology, which urgently needs to be addressed by researchers. Finally, most LR-DDoS attack defense methods only focus on detecting attacks and do not consider effective real-time mitigation of attack traffic.

To overcome the above problems, we propose an encrypted LR-DDoS attack detection and mitigation method based on multi-granularity feature fusion (MFFLR-DDoS). Initially, the MFFLR-DDoS model analyzed the encrypted session from the temporal nature of the packet and the

spatial nature of the session and used the gated recurrent unit (GRU) [3] to extract the timing features of the packet. A hybrid model of capsule network (CapsNet) [4] and multi-head self-attention (Muiti-head Attention) [5] is used to extract the spatial features of the session. Then, the MFFLR-DDoS model fuses the extracted two types of features and inputs them into the sigmoid classifier for abnormal traffic identification. Finally, taking advantage of the real-time control of SDN architecture, we used the SDN controller to issue flow rules to defend against attack behaviors in real-time. The experimental results show that the MFFLR-DDoS model has higher detection accuracy and lower false alarm rate than advanced methods, and can effectively alleviate attack traffic in a short time. The main contributions of this paper are as follows.

- The previous LR-DDoS attack detection methods did not consider encryption. We analyze encrypted LR-DDoS attacks from two aspects: LR-DDoS attack behavior and transport layer security protocol (TLS).
- To effectively detect encrypted LR-DDoS attacks, this paper analyzes the encrypted session flow from the time sequence of data packets and the spatiality of the session. The GRU model is used to extract the timing features between packets, the CapsNet model is used to extract the spatiality detail features of the session, and the multi-head attention mechanism is used to filter out the key information. The encrypted LR-DDoS attacks are detected from different granularities, which effectively improves the detection accuracy of LR-DDoS attacks and reduces the false positive rate.
- We propose an online real-time mitigation strategy for LR-DDoS attacks based on OpenFlow technology.

The rest of this paper is structured as follows. Section 2 presents the related work. Section 3 elaborates on the proposed model. Section 4 is devoted to the experimental results and analysis. Section 5 concludes the paper and looks to the future.

2. Related work

2.1. DDoS attack detection method in SDN

SDN network architecture shields the hardware differences of the underlying devices, simplifies the configuration and management of network devices, and realizes the flexible deployment of network devices; thus, it has attracted much attention from academia and industry. However, SDN introduces many security threats, among which DDoS attack is the most harmful. At present, there are many research results on DDoS attack detection in the field of SDN. Liu et al. [6] proposed a two-stage DDoS attack detection scheme. In the first stage, the information entropy mechanism was used to coarse-grained detect suspicious components and ports. In the second stage, CNN was used to fine-grained detect normal flow and abnormal flow. Finally, the SDN controller executes the defense strategy to receive the attack flow. Experimental results show that the detection accuracy of this method reaches 98.98%, and it is suspicious and effective in detecting DDoS attacks in SDN. Bhayo et al. [7] proposed a new SDN-based security framework that can detect vulnerabilities in IoT devices or malicious traffic generated by IoT devices through session IP counters and IP payloads, which detects DDoS attacks at an early stage with high accuracy and low false positive rate. Cao et al. [8] proposed a detection method based on spatio-temporal graph convolution (ST-GCN), which extracts effective features from the temporal and spatial distribution of network traffic and performs anomaly detection. Experimental results show that the proposed method can detect DDoS attacks accurately. Compared with the classical detection method, the detection accuracy is improved by

nearly 10%. Zhang et al. [9] proposed a two-stage abnormal traffic detection method for DDoS attack detection in SDN. In the first stage, the method of information entropy is used to make coarse-grained judgments of abnormal traffic, and in the second stage, the deep learning hybrid model SSAE-SVM is used to make fine-grained judgments of abnormal traffic. The experimental results show that this method can identify more than 98% of DDoS traffic, and has good detection performance. Wang et al. [10] proposed a real-time DDoS attack detection method, which used the improved firefly algorithm to optimize the convolutional neural network (CNN) and detect DDoS attacks in the SDN environment. Experimental results show that the method can effectively identify DDoS attacks. Chauhan et al. [11] used random forest (RF), light gradient boosting machine (LGBM), support vector machine (SVM), and k-nearest neighbor (KNN)methods to train and test DDoS attacks and normal data. The experimental results show that the LGBM model has the highest detection accuracy and can detect DDoS attacks in real-time.

2.2. LR-DDoS attack detection method

Low Rate Denial of Service attack against HTTP protocol is a new attack method in recent years. This kind of attack only needs a small number of hosts and occupies the link resources of the server at a low broadband rate to achieve the purpose of the attack. Due to the low cost and remarkable effect of LR-DDoS attacks, it has been frequently used to attack Internet sites in recent years, which has caused great hidden dangers to network security. Therefore, it has received the attention of some scholars and experts. Gogoi et al. [12] proposed an LR-DDoS attack detection method based on long short-term memory (LSTM) deep learning. The study is verified on the CICDoS dataset and a synthetically generated dataset, and the accuracy reaches 99%. Nugraha et al. [13] proposed a deep learning hybrid model CNN-LSTM to detect LR-DDoS attacks in SDN. The results show that the accuracy of CNN-LSTM is 99%, which exceeds the methods such as MLP and SVM. Muraleedharan et al. [14] used the deep learning model to detect LR-DDoS attack traffic, and the experimental results show that the classifier can obtain an accuracy of 99.61%. Xu et al. [15] proposed an LR-DDoS attack detection method combining the one-dimensional convolutional neural network and gated recurrent unit based on a hybrid deep neural network. The average detection rate of this method is 98.68%, which is more advantageous than MF-DFA or power spectral density-based detection methods. Chen et al. [16] used a one-dimensional CNN model to detect HTTP slow denial of service attack traffic. The model extracted sequence segments through convolution kernels, and learned the local patterns of attack samples from the segments, so that the model had the ability to detect data flows with multiple attack frequencies. The results show that the detection accuracy and accuracy of the model reach 96.76% and 94.13% respectively, which also has a good detection ability for unknown attacks. In addition, some scholars have considered the LR-DDoS attack defense method under the new network SDN. For example, Wang et al. [17] proposed a trustworthiness-based LR-DDoS attack countermeasure (CCSA) using SDN technology, which evaluates each client by its connection and frequency of sending fragmentation requests. Connections to low-reputation clients will be blocked to avoid them consuming resources. When the server resource is insufficient, the suspicious connection will be suspended to ensure the availability of the server. Simulation results show that CCSA can effectively prevent LR-DDoS attacks and maintain low memory utilization of the controller. Yungaicela-Naula et al. [18] proposed a deep reinforcement learning based on SDN network architecture for attack detection and defense against LR-DDoS attacks. Experiments show that the average detection rate of this method reaches 98%, and the success rate of mitigating LR-DDoS attacks in time for a few attackers reaches 100%.

2.3. Malicious encrypted traffic detection method

Nowadays, network activity is becoming more and more frequent, and the encrypted traffic is also increasing. Some attackers use encryption to hide their attack content, to avoid the existing detection methods, resulting in a high false detection rate. Therefore, encrypted malicious traffic detection has become a problem that cannot be ignored today. In recent years, many experts and scholars have begun to study malicious traffic encryption technology. There are mainly two detection modes based on statistical features and original traffic feature input. The method based on statistical features is the most commonly used method for traffic identification. This method directly extracts or calculates the statistical values of sample attributes as input features, and then uses machine learning or deep learning to classify the input statistical features. Li et al. [19] extracted features from various aspects such as TLS cipher suite usage, certificate, and domain name during the handshake phase, and used RF method to detect encrypted flow, to find malicious encrypted traffic. Ferriyan et al. [20] proposed the TLS2Vec method, which converts the features in TLS handshake and payload into vectors through Word2Vec and inputs them into LSTM and Bi-LSTM for encrypted malicious traffic identification. Gu et al. [21] used plaintext information in data packets and handshake packets as features and used the MGREL method to detect encrypted malicious traffic. Gracia et al. [22] took TCP window size, TCP flag, TLS content type, TLS record, and other fields as features and input them into the distributed clustering model deployed in Apache Spark to realize the detection of encrypted LR-DDoS attacks. However, there are two problems in the detection method based on statistical features. Initially, the feature extraction method relies too much on expert experience and is greatly affected by unknown features, which is easy to cause a high false positive rate. Second, the extracted features are limited and only for specialized attack patterns. Furthermore, some experts and scholars use raw traffic data as input to study malicious encrypted traffic detection. Specifically, Tang et al. [23] proposed selecting the first 784 bytes of raw traffic as model input and then using CapsNet and LSTM to learn the time series and spatial features of encrypted traffic, effectively identifying malicious encrypted traffic. Lotfollahi et al. [24] used IP headers and the first 1480 bytes of IP packet payloads as inputs to CNN and SAE models to achieve malicious encrypted traffic classification. Cui et al. [25] extracted 784 bytes from the encrypted flow header as model input and then used CapsNet to learn the spatial and byte features of the session flow. Zou et al. [26] combined CNN and LSTM to learn intra-packet features of the first 784 bytes in a single data packet and used LSTM to learn inter-packet features of any three consecutive data packets. In addition, other similar studies [27,28] also achieved high detection rates. However, these studies using the payload of encrypted traffic as an input feature may lead to privacy risks. Second, the large amount of encrypted payload data brings enormous pressure to model training, which cannot guarantee the real-time performance of the detection system.

The above works have proposed network attack detection techniques, but there are few studies on encrypted LR-DDoS attack detection methods. However, with the wide application of encryption technology in Internet communication, it is urgent to improve the detection technology of encrypted traffic of the LR-DDoS attack. In this paper, we propose an encrypted LR-DDoS attack recognition method based on multi-granularity feature fusion in the SDN network. This method analyzes the encrypted session flow from different perspectives, and uses deep-learning for feature extraction, to improve the detection efficiency. Moreover, utilizing the architectural advantages of SDN, an effective strategy for mitigating malicious traffic has been proposed.

3. System design

3.1. System overview

As a new type of network architecture, SDN's core design concept separates the data forwarding and decision control functions of network devices, achieving centralized hardware control. This separation technology improves the openness and programmability of the network, achieving its scalability. Developers can flexibly use high-level languages to design applications and modify network strategies on the upper layer without considering the underlying hardware. Similarly, this advantage simplifies the deployment of online attack detection and mitigation solutions. Based on the advantages of SDN mentioned above, we propose an encrypted LR-DDoS attack detection framework for the web application layer, as shown in Figure 1. The architecture design has modularity, flexibility, and scalability. It consists of three modules: data preprocessing module, anomaly detection module, and mitigation module. (1) Data preprocessing module: Initially, use the Wireshark application to collect traffic and forward it to the data preprocessing module for data processing. Then, use the 5-tuple (source IP, destination IP, source port, destination port, and transport layer protocol) as the basis for session partitioning, and use the SplitCap [29] tool to partition the network flow into session forms. Finally, forward the session samples to the anomaly detection module for attack identification. (2) Anomaly detection module: Initially, the MFFLR-DDoS model is used to extract the features of the preprocessed traffic from two aspects: the spatiality features of the session and the timing features of the packet. Then, deep learning technology was used to identify LR-DDoS attacks. Finally, the attack flow is saved to the blacklist. (3) Mitigation module: Based on the detection module results, issue defense measures and prevent attackers from invading the system again. We will provide a detailed introduction to the specific implementation of each module in the remaining sections of Section 3.

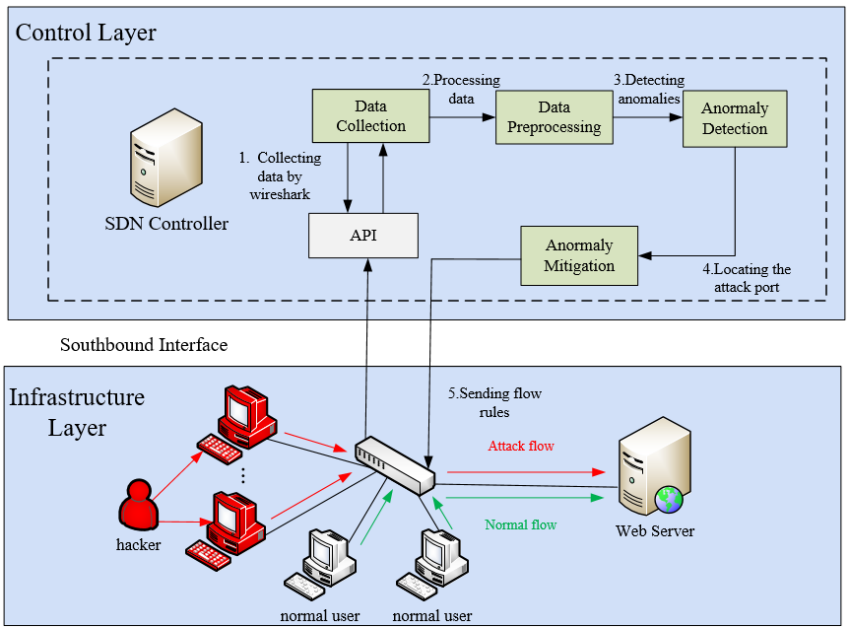


Figure 1. SDN-based LR-DDoS attack detection framework.

3.2. Data preprocessing

The data packet is the basic transmission unit of network flow. Each data packet consists of a fixed format of the packet header and payload. Therefore, network flow has a hierarchical structure of "flow-packet-byte". Compared with one-way flow, from the perspective of traffic communication interaction, the session composed of data packets from both sides of communication contains more interactive information. This paper chooses the session as the flow classification granularity, and the session representation is shown in Eq (1):

$$Packet = \{packet_1 = (tup_1, len_1, tim_1), packet_2 = (tup_2, len_2, tim_2), ..., packet_n = (tup_n, len_n, tim_n)\}$$
$$Session = (tup, len, tim, dur) \tag{1}$$

where $packet_i$ is the data packet sorted by time; tup is the interchangeable five-tuple information between the source and destination, denoted as $tup = (IP_S, IP_D, Port_S, Port_D, \Pr otocol)$. $len = \sum_{1}^{n} len_i$ is the sum of the length of each packet in the session. tim_i represents the start time of the i-th packet, arrange these packets in chronological order, $tim_1 < ... < tim_n$; $dur = tim_n - tim_1$ is the duration of the session. Therefore, the original traffic can be expressed as $Flow = \{Session_1, Session_2, \cdots, Session_m\}$ and m is the total number of session flows.

3.3. Multi-feature fusion intrusion detection model

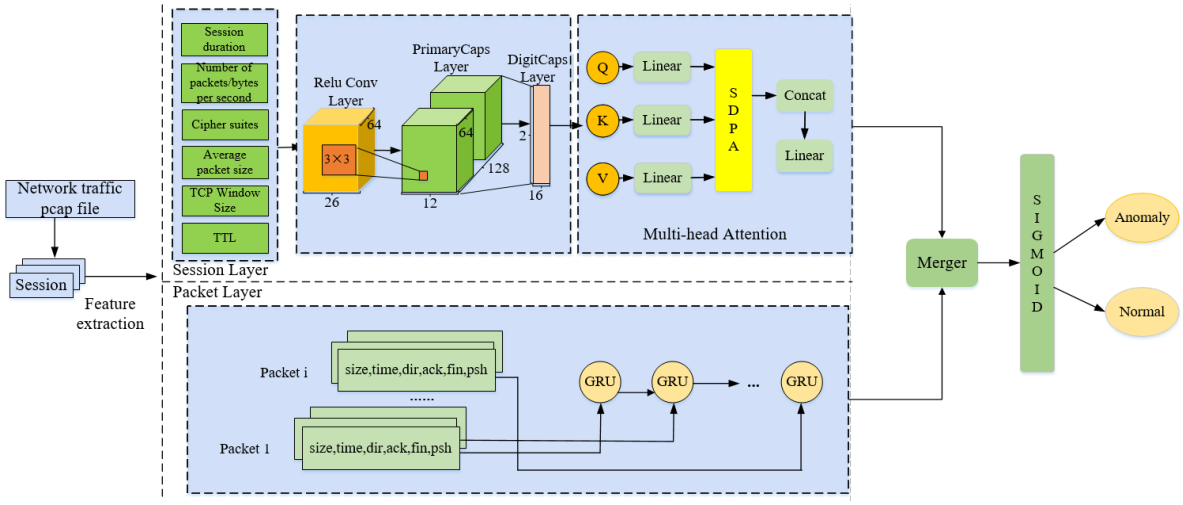


Figure 2. Encrypted LR-DDoS attack detection framework based on MFFLR-DDoS.

The LR-DDoS attack detection framework based on multi-granularity feature fusion encryption is shown in Figure 2, which includes three stages: Data preprocessing, multi-granularity feature extraction, and attack classification. The multi-granularity feature extraction is analyzed from both the packet and session levels. Regarding data packets, the GRU model is used to extract the time series features of the first n data packets of each session (to ensure consistency in data processing, the first n data packets of the session are taken). In terms of the session, we first extract local statistical features from TLS packets in the session and use CapsNet to extract spatial details of the

session. Then, we use the multi-head attention mechanism to extract key statistical features in the session. Finally, the features extracted from multi-granularity are fused and input into the sigmoid function for LR-DDoS attack judgment.

3.3.1. Feature extraction of packet timing based on GRU

When an LR-DDoS attack occurs, data packets are transmitted to the server at a very slow rate and the payload content of the transmitted data packets is relatively small. Therefore, the load size and time interval of data packets can effectively distinguish malicious session samples. In addition, the encrypted LR-DDoS attack uses TCP handshake protocol when transmitting data packets, resulting in a strong correlation between the data packet and the transmission direction during transmission. To explore the global communication behavior of traffic, we divide the session flow at packet granularity. According to statistics, the majority of sample packets have fewer than 30, so the first 30 packets of the flow are selected to represent global flows. The representation of the session at packet granularity is as follows:

$$session_{\text{pack}} = \{p_1, p_2, \cdots, p_{30}\} \tag{2}$$

$$p_t = \{size, time, dir, ack, fin, psh\} \tag{3}$$

In the t-th packet p_t, $size$ represents the payload size of the packet, $time$ represents the interval of one packet at the distance, dir represents the transmission direction, and ack, fin, and psh are the flags of the TCP layer. The packet's payload size, time interval, transmission direction, and TCP flag bits represent the timing state of the traffic.

This article uses the GRU model to learn the global interaction of session flow, mining the temporal relationships between data packets, and obtaining the global behavior representation of the session. GRU has made improvements based on LSTM by merging the input gate and the forget gate into the update gate z_t, which is used to determine how to combine past information with current input information and control the degree of retention of old information. In addition, GRU also has a gate called reset gate r_t, which is used to determine whether to ignore past states. Its function is to control the degree to which past states have an impact on the current state. The reset gate can help the model capture long-term dependencies to some extent. Each of the aforementioned data packets p_t corresponds to one input of the GRU. The specific calculation formula is as follows:

$$z_t = \sigma(W_z p_t + U_z h_{t-1} + b_z) \tag{4}$$

$$r_t = \sigma(W_r p_t + U_r h_{t-1} + b_r) \tag{5}$$

$$\tilde{h}_t = \tanh(W_h p_t + U_h(r_t \otimes h_{t-1}) + b_h) \tag{6}$$

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \tilde{h}_t \tag{7}$$

$$temp\_feature = GRU(p_1, p_2, \cdots, p_{30}) \tag{8}$$

In Eqs (4)–(8), p_t and h_t respectively represent the input and output vectors at the current time, and the corresponding node h_t is obtained by calculating the p_t at each time. h_{t-1} is the output

vector at time $t-1$, $\tilde{h}_t$ is the candidate activation state, $W_z, W_r, W_h, U_z, U_r, U_h$ are the weight matrices, b_z, b_r and b_h are biased parameters, $\sigma(\bullet)$ and $\tanh(\bullet)$ are sigmoid functions and hyperbolic tangent functions, respectively, $\otimes$ represents Hadamard product. If z_t approaches 1, it indicates that more current state information can be transmitted; On the contrary, it indicates that the current state information is transmitted very little. If r_t approaches 1, it indicates that the current state retains more past information; On the contrary, it forgets more past information.

3.3.2. Session statistical feature extraction based on CapsNet and Multi-head attention

In the session layer, to effectively extract encrypted LR-DDoS attack features, we select features for encrypted malicious traffic detection from the following six aspects according to the paper [22]. For example: 1). Session duration. 2). The number of packets and bytes sent per second. 3). TCP window size. 4). Time to Live (TTL). 5). The size of the packet. 6). Cipher suites. The statistical characteristics based on the session layer are shown in Table 1 and described as follows:

(1) Session duration. Due to LR-DDoS attacks injecting low-capacity legitimate traffic at a very slow speed to attack application and server resources. Therefore, the session duration during the attack is longer than normal.

(2) The number of packets and bytes sent per second. Due to the characteristics of Slowloris and Slow body attacks, data is sent to the server at a very slow speed, occupying bandwidth resources, and preventing legitimate users from accessing it normally. Therefore, the number of data packets sent per unit time is very small. Similarly, the number of bytes sent per unit time is also very small.

(3) TCP window size. When a Slow read attack occurs, when the server returns data to the client, the attacker usually sets a very small buffer to the client, causing the client to read the response at a very low rate. If the client fails to read data for a long time, it will send a TCP Zero Window to the server, causing the server to mistakenly assume that the client is busy and thus continuously occupying the thread. Therefore, the TCP reception window of the client can be used to determine LR-DDoS attacks.

(4) Time to live. Because attackers typically launch attacks on the server side along a fixed route. However, when normal users access the server, the route is random, so time to live can be an important basis for distinguishing between LR-DDoS attacks.

(5) The size of the data packet. Due to the attacker sending data packets to occupy bandwidth resources rather than transmit data, the packet load size is relatively small. When legitimate users transmit data, the packet load size is random, so the packet load size can distinguish between LR-DDoS attacks.

(6) Encryption suite. Encryption suite refers to the combination of encryption algorithms used by servers and clients in SSL/TLS communication. Attackers typically use a simple and limited number of encryption suites. However, the encryption suite for normal users has diversity. Therefore, encryption suites can distinguish between malicious encrypted traffic.

Table 1. Traffic statistical features.

No.	Name	No.	Name
1	Session duration	14	Average TTL in Uplink
2	Number of packets sent in 1s in Uplink	15	Maximal TTL in Downlink
3	Number of packets sent in 1s in Downlink	16	Minimal TTL in Downlink
4	Number of bytes sent in 1s in Uplink	17	Average TTL in Downlink
5	Number of bytes sent in 1s in Downlink	18	The number of cipher suites provided by the client
6	Maximal size of TCP window in Uplink	19	The number of cipher suites selected by the server
7	Minimal size of TCP window in Uplink	20	Extend the number of cipher suites
8	Average size of TCP window in Uplink	21	Maximal packet size in Uplink
9	Maximal size of TCP window in Downlink	22	Minimal packet size in Uplink
10	Minimal size of TCP window in Downlink	23	Average packet size in Uplink
11	Average size of TCP window in Downlink	24	Maximal packet size in Downlink
12	Maximal TTL in Uplink	25	Minimal packet size in Downlink
13	Minimal TTL in Uplink	26	Average packet size in Downlink

1) Capsule network

To effectively extract the spatial detail features of the session, we adopt the method of CapsNet. The capsule network is mainly composed of the convolutional layer, the primary capsule layer, and the digital capsule layer, as shown in Figure 3. Compared to traditional scalar neurons, the capsule network changes the input and output of neurons to vector structure, marking important features in the data through vectors. The direction of the vector represents the probability of being in that direction, and the length of the vector represents the magnitude of the probability. The parameters between capsule layers are iteratively updated through dynamic routing algorithms, and the data features extracted from the primary capsule layer are transmitted to the digital capsule layer for classification calculation. Due to the use of "dynamic routing" in CapsNet, which utilizes a weight matrix, single-layer neurons only focus on the most active feature selectors in the local pool, while ignoring other selectors with less correlation, resulting in higher accuracy of the classifier.

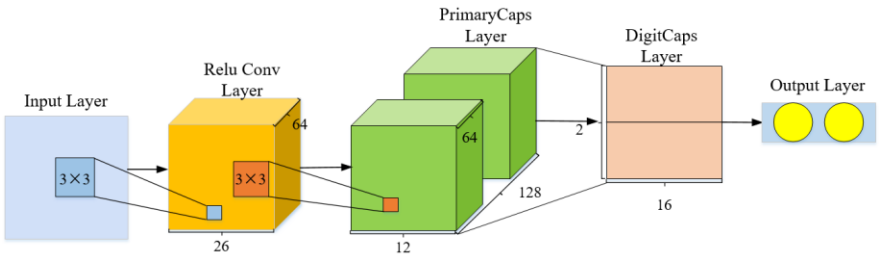


Figure 3. Capsule network architecture.

The steps of the capsule network algorithm are as follows:

Step 1: Initialize the weight matrix w_{ij}, the input vector is x_i, and then the prediction vector $x_{j|i}$ can be expressed as:

$$x_{j|i} = w_{ij} x_i \tag{9}$$

Step 2: The softmax function is used to calculate the coupling coefficient c_{ij} by updating the intermediate variable b_{ij}, and the deep feature vector s_j is obtained by weighted sum with the prediction vector $x_{j|i}$. The calculation formula is as follows:

$$c_{ij} = soft\max(b_{ij}) = \frac{\exp(b_{ij})}{\sum_{1\le k\le j}\exp(b_{ij})} \tag{10}$$

$$s_j = \sum_i c_{ij} x_{j|i} \tag{11}$$

Step 3: Use the squash activation function to compress the deep feature vector s_j, and compress the modulus length of the vector to between 0–1 to obtain the output vector v_j:

$$v_j = squash(s_j) = \frac{\|s_j\|^2}{0.5+\|s_j\|^2} \bullet \frac{s_j}{\|s_j\|} \tag{12}$$

Step 4: Update the intermediate variable b_{ij} by calculating the correlation between the prediction vector $x_{j|i}$ and v_j through the inner product, the formula is as follows:

$$b_{ij} \leftarrow b_{ij} + x_{j|i} v_j \tag{13}$$

Step 5: Update the coupling coefficient c_{ij} according to Eq (10); Subsequently, s_j and v_j are updated in turn through Eqs (11) and (12). The dynamic routing algorithm obtains the best output vector v_j after r iterations, whose direction represents a certain class of features, and whose modulus length represents the probability p_j of the class:

$$p_j = \|v_j\| \tag{14}$$

2) Multi-head attention

The attention mechanism can selectively focus on key information in the data. To further improve the detection rate of encrypted LR-DDoS attacks, we introduce the Multi-head Attention mechanism (MHA) to filter out the less relevant statistical features extracted in the previous stage and retain the important information in the session. The essence of the multi-head attention mechanism is the combination of multiple self-attention structures. In this paper, we adopt scaled dot-product attention (SDPA) to mine key statistical features using query $Q = W_Q X$, key $K = W_k X$, and value $V = W_V X$.

$$SDPA(Q,K,V) = soft\max\left(\frac{QK^T}{\sqrt{d}}\right)X \tag{15}$$

Where X is a vector matrix of model inputs, where each row represents all statistical features in a session; Matrices W_Q, W_k, and W_v are learnable model parameters. d is the size of the hidden layer of the network and also the dimension of the Q and K vectors. The scaling operation is performed so that the parameters of the softmax function do not become too large when using larger key sizes.

In the Multi-head attention mechanism, to learn the relevant information of the session flow

from different dimensions, it is necessary to do a linear transformation of the vector Q, matrix K, and matrix V using different parameters multiple times, and input the results into the SDPA to obtain different attention outputs. Then the attention output $head_i$ of each head can be expressed as follows.

$$head_i = SDPA\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{16}$$

where W_i^Q, W_i^K, and W_i^V are linear transformation parameters. Linear transformations over the different parameters of Q, K, and V are the essence of "multi-head".

After computing the scaled dot product attention n times, the output weight matrix W^o is used to connect the n parallel heads $head_1$ to $head_n$, and the obtained value is the result of MHA.

$$MHA(Q,K,V) = concat[head_1,...,head_n]W^o \tag{17}$$

In general, the self-attention mechanism uses multiple heads to analyze the statistical features of the session flow from multiple perspectives, so that more comprehensive and accurate session features can be extracted.

Finally, we extract the spatial features of the session using the CapsNet-Multihead method, and the Eq (18) is as follows:

$$spatio\_feature = CapsNet - Multihead\left(f_1, f_2, \cdots, f_{26}\right) \tag{18}$$

Where f_i represents the statistical characteristics of the session.

We divide the traffic into packet granularity and session granularity to obtain session-based spatial features $spatio\_feature$ and packet-based temporal features $temp\_feature$. Then, we merge the features at both granularities and output the discrimination results through the Sigmoid function, as shown in Eq (19).

$$output = Sigmoid(concatenate[temp\_feature, spatio\_feature]) \tag{19}$$

3.4. Mitigation strategies

According to the above attack detection, this article proposes a defense method based on the advantage of SDN controllers being able to control the network in real-time. Once the attack is detected, the SDN controller based on the OpenFlow protocol [30] can defend against the attack behavior in real-time by issuing flow rules. Initially, the SDN controller locates the switch port to which the attack host is connected according to the flow table information extracted from the attack flow; Then, isolates the attack hosts by issuing flow rules. This method can permanently offline the attacked host of the network, effectively preventing attackers from tampering with IP or Mac addresses and injecting attack flow into the network again. The specific defense process is as follows:

Step 1: Extract the information. Once the attack traffic is detected, the source IP (src_ip), destination IP (dst_ip), switch ID (s_id), switch input port (in_port), timestamp (time_stamp), and other information are extracted from the flow table and put into the blacklist library in the SDN controller.

Step 2: Locate the attack port. First, group the flow table information in the blacklist based on the src_ip and dst_ip fields, and sort each group of information in descending order according to the time_stamp field. Then, find the flow table information with the minimum time_stamp value, and

based on the s_id and in_ port locate the attacked host. If the input port of the switch is connected to the host, proceed to the next step; Otherwise, the controller notifies the neighborhood controller through the east-west interface and searches for the same src_ip, dst_ip, and other flow table information in the controller's neighborhood, and locates the attack port according to the time_stamp.

Step 3: Flow rules generation. The flow rules shown in Table 2 are constructed with the matching field of in_port = the port connected to the attack host & ipv4_src = source IP, and Action = Drop.

Step 4: Flow rules issued. The SDN controller drops the attack flow by issuing flow rules (Table 2) to achieve the LR-DDoS attack defense.

Table 2. Flow rule.

Match	Priority	Action
in_port:the port connected to the attack host ipv4_src: source IP	Very high	Drop

4. Experiment and analysis

4.1. Experimental environment and model parameter setting

This article uses the Mininet [31] network simulator to simulate a real software defined networking environment. All software is deployed on the operating system Ubuntu 18.04 TLS, and this experiment supports the OpenFlow protocol. The controller and switch use the open-source RYU controller and the OpenvSwitch switch in Mininet, respectively. Figure 4 shows the SDN network topology generated in Mininet, which we used to validate the proposed LR-DDoS attack detection mechanism. The SDN network contains 5 OpenFlow switches S1~S5, with 12 hosts h1~h12, of which h9 is a web server. We chose five hosts, h1~ h5, connected to S1 and S2, as the attack hosts, and launched LR-DDoS attacks on the victim host h9 using the Slowhttptest tool [32] (including Slow Header, Slow Post, and Slow Read), and generated network background traffic using the D-ITG tool [33].

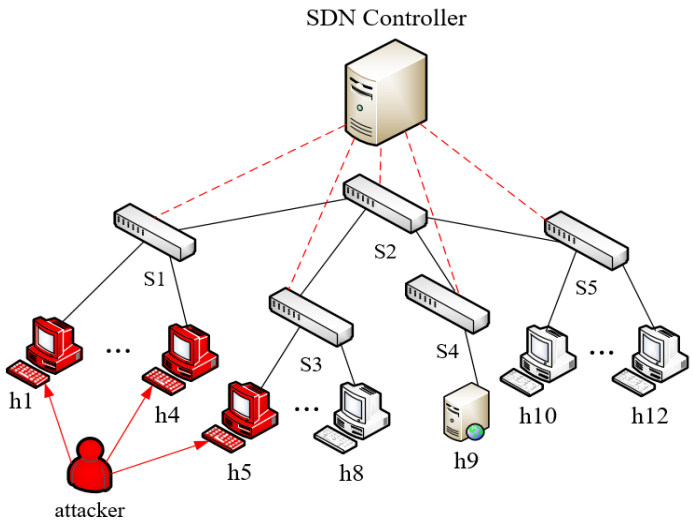


Figure 4. Experimental topology.

The parameter Settings of the MFFLR-DDoS model are shown in Table 3 and described as follows: The capsule network consists of the convolutional layer, the primary capsule layer, and the digital capsule layer. In the convolutional layer, the number of convolution kernels is 64, the kernel size is 3×3, the step size is 1, and the activation function is RELU. In the primary capsule layer, the number of convolution kernel groups is 2, the number of convolution kernels in each group is 64, the size of convolution kernels is 3×3, the step size is 2, and the activation function is RELU. In the digital capsule layer, the number of output capsules is 2, and the dimension of each capsule is 16. The number of parallel heads in the multi-head attention layer is 16. There are two hidden layers in the GRU model, the number of neurons in the first layer is 8, and the number of neurons in the second layer is 16. In the Dropout layer, the forgetting rate is 0.1. Finally, the concatenate function is used to fuse the features extracted by each model. The output layer uses the sigmoid function for classification, the loss function uses binary_crossentropy, and the optimization algorithm uses Adam, batch_size is set to 25, and epoch is 30. The ratio of training set, validation set, and test set is set to 8:1:1, respectively.

Table 3. Parameter settings of MFFLR-DDoS model.

Network Structure	Layer	Output shape
CapsNet+Multi-head Attention	InputLayer	(None,25)
	Dense	(None,784)
	Reshape	(None,28,28,1)
	Conv2D	(None,26,26,64)
	PrimaryCaps	(None,12,12,128)
	Reshape	(None,144,128)
	DigitCaps	(None,2,16)
	Muliti-head attention	(None,2,16)
	Flatten1	(None,32)
GRU	InputLayer	(None,1,5)
	GRU1	(None,1,8)
	Dropout	(None,1,8)
	GRU2	(None,1,16)
	Dropout	(None,1,16)
	Flatten2	(None,16）
Fussion	Concatenate	(None,48)
	Dense	(None,1)

4.2. Datasets

1) SlowDDoS: Encryption SlowDDoS attack detection is an important research task in DDoS attack detection. However, there is currently little work on building datasets for low-rate DDoS attacks. To meet this requirement, we collected a dataset called SlowDDoS using the network topology shown in Figure 4, which consists of attack and normal samples. We collected a total of 20830 samples, including 10312 attack samples and 10518 normal samples.

2) ISCX-2016-SlowDoS [34]: the ISCX-2016-SlowDoS dataset is a public dataset about application layer DoS attacks. This dataset includes both high-capacity and low-capacity application layer DoS attacks. Since the proposed model focuses on low-speed DoS attacks, we extract

low-volume DoS attack data from the ISCX-2016-SlowDoS dataset as attack samples. We use Monday's data from the CIC-IDS2017 dataset as normal samples. We collected a total of 17270 samples, including 8803 attack samples and 8467 normal samples.

4.3. Evaluation metric

To effectively evaluate the performance of the proposed MFFLR-DDoS model, we use six metrics: Accuracy (Acc), precision (P), recall (R), F1-score(F1), detection rate (DR), and false alarm rate (FAR) for evaluation, as shown in Eqs (20)–(24). True positive (TP): The number of samples in the attack class is classified as the attack. False positive (FP): Classify the number of samples in the normal class as the attack; true negative (TN): The number of samples in the normal class is classified as normal. False negative (FN): The number of samples in the attack class is classified as normal.

$$Acc = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{20}$$

$$P = \frac{TP}{TP + FP} \times 100\% \tag{21}$$

$$R = DR = \frac{TP}{TP + FN} \times 100\% \tag{22}$$

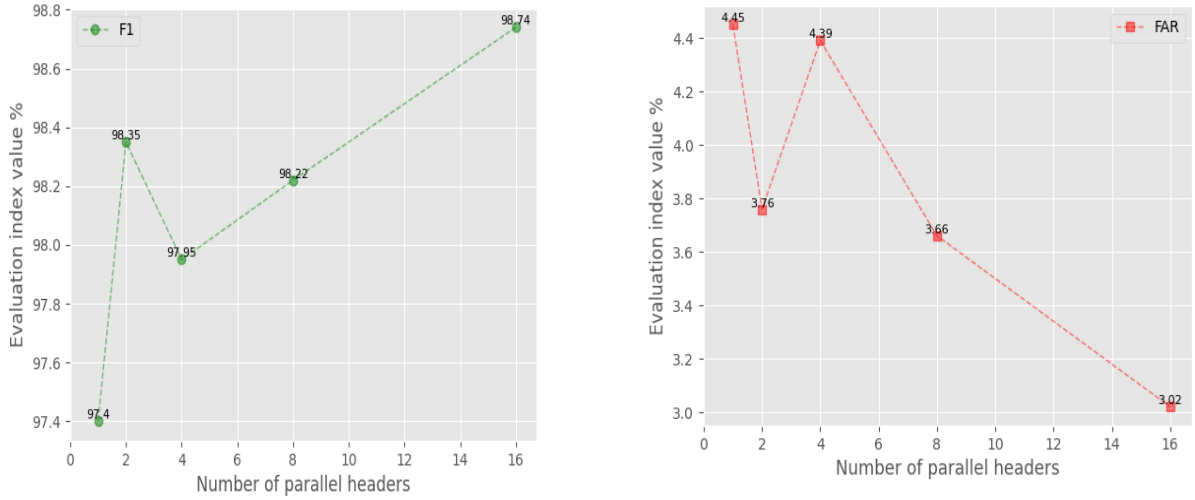
$$F_1 = 2 \times \frac{P \times R}{P + R} \times 100\% \tag{23}$$

$$FAR = \frac{FP}{FP + TN} \times 100\% \tag{24}$$

4.4. Experimental results

4.4.1. Parameter selection for Multi-head attention models

In the multi-head attention layer, the number of hyperparameter parallel heads will affect the attention degree of different features. The appropriate number of parallel heads can more accurately extract the key features in the session, and too many or too few parallel heads may cause the loss or interference of effective features, thus affecting the classification effect of the model. In this paper, the number of parallel heads is set as 1, 2, 4, 8, and 16, respectively, and the experimental results are shown in Figure 5. It can be seen that when the number of parallel headers is 16, the F1-score and false alarm rate are better than the other number of parallel headers. When the number of parallel heads is 1, that is, single-head self-attention, the mining ability of the MFFLR-DDoS model for multi-dimensional feature information is worse than that of multi-head attention.



(a) F1-score (b) FAR

Figure 5. Influence of the number of parallel heads on the classification effect of the model.

4.4.2. Ablation experiment

In order to verify the effectiveness of each part of the model, we designed ablation experiments from the following two aspects. 1. The fusion effect of different granularity and 2. The effect of different session-based representation methods. The experiments are carried out on the SlowDDoS datasets.

1) The fusion effect of different granularity

- w/o packet granularity: Based on the MFFLR-DDoS model, the packet-based feature extraction layer (GRU model) is removed.
- w/o session granularity: Based on the MFFLR-DDoS model, the session-based feature extraction layer (CapsNet and Multi-head Attention model) is removed.

Table 4. Ablation experiments with different granularities.

Method	Accuracy	Precision	Recall	F1-score
w/o packet granularity	97.48%	96.62%	99.03%	97.81%
w/o session granularity	78.77%	73.14%	98.95%	84.11%
Our method	98.69%	98.75%	98.95%	98.85%

From Table 4, it can be seen that the feature extraction method based on session granularity has better detection performance compared to the feature extraction method based on packet granularity. After adding the feature extraction method based on packet granularity, the accuracy is improved, which indicates that the precision of malicious flow detection is improved, that is, the false positive samples are reduced. The recall rate decreases, indicating that the recall rate for malicious flows decreases, that is, the missed samples increase. The reason why the detection effect is improved after the fusion of the two feature extraction methods is that the feature extraction methods based on the two granularities can describe the encrypted LR-DDoS attacks from different perspectives, and can make up for the shortcomings of insufficient feature extraction of the single granularity. In addition,

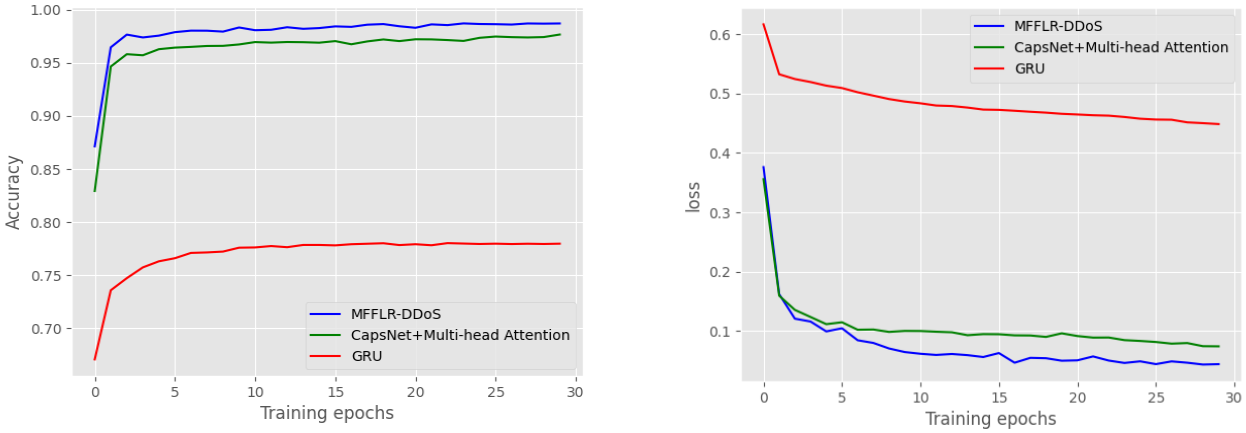
when detecting the application of the model, enterprises paying more attention to the model can have a low false positive rate, under the premise of Improving the F1-score, it is worth sacrificing the recall rate appropriately.

2) Different representation learning models based on session granularity

Table 5. Ablation experiments for different models.

Method	Accuracy	Precision	Recall	F1-score
CapsNet	96.40%	96.70%	96.97%	96.84%
CNN(1D)	92.03%	88.26%	99.15%	93.39%
LSTM	78.21%	72.45%	99.43%	83.82%
GRU	78.77%	73.14%	98.95%	84.11%
CapsNet+Multi-head Attention	96.59%	97.47%	96.49%	96.98%

As can be seen from Table 5, this paper attempts a variety of representational learning models based on conversational granularity. Among CapsNet, LSTM, CNN (1D), GRU, and CapsNet+Multi-head Attention, CapsNet+Multi-head Attention has better performance. Thanks to the spatial characteristics of flow, the spatial model can better capture the detailed information flow. The CapsNet with Multi-head Attention is better than the pure CapsNet, which proves that the introduction of attention improves the detection effect of the model to a certain extent by mining the relationship between the statistical features of each session flow.



(a) Accuracy degree (b) Loss degree

Figure 6. Comparison of classification accuracy and loss degree for different granularity model.

To demonstrate the superiority of the proposed MFFLR-DDoS model, we compared it with the CapsNet+Muliti-head Attention and GRU models on the SlowDDoS dataset. From Figure 6(a), it can be seen that the training speed of the MFFLR-DDoS model is faster than that of the CapsNet+Muliti-head Attention and GRU models. At the fifth round of model training, the accuracy begins to converge and approaches 98%, which is higher than the accuracy of other models. This indicates that the MFFLR-DDoS model has strong feature extraction ability and can quickly reach the fitting state. From Figure 6(b), it can be seen that the loss rate of the MFFLR-DDoS model

quickly decreases to close to 0, and the loss value tends to stabilize and no longer changes at the 10th iteration. However, the GRU model and the CpasNet+Multi-head Attention model have a slower decrease in loss rate and poor convergence. It can be seen that the MFFLR-DDoS model has high classification accuracy and good robustness.

We used ROC and AUC metrics to evaluate the performance of three classification models: MFFLR-DDoS, CapsNet+Muliti-head Attention, and GRU. The Receiver Operating Characteristic Curve (ROC) is a graphical tool that depicts the performance of a classifier, displaying the relationship between the true positive rate (TPR) and false positive rate (FPR) of the classifier at different thresholds. The area under the curve (AUC) value represents the area under the ROC curve and is used to measure the performance of the classifier. The closer the AUC value is to 1, the better the performance of the classifier; on the contrary, if the AUC value approaches 0, it indicates poorer classifier performance. From Figure 7, it can be seen that the ROC curves of MFFLR-DDoS and CapsNet+Muliti-head Attention are steeper, indicating that the performance of MFFLR-DDoS and CapsNet+Muliti-head Attention models is relatively good. The AUC value of MFFLR-DDoS is 0.986, the AUC value of CapsNet+Muliti-head Attention is 0.972, and the AUC value of GRU is 0.758. The AUC value of MFFLR-DDoS is higher than the other two classification algorithms, indicating that the MFFLR-DDoS model has better performance in traffic detection. In summary, the performance of our proposed MFFLR-DDoS model is superior to other models.

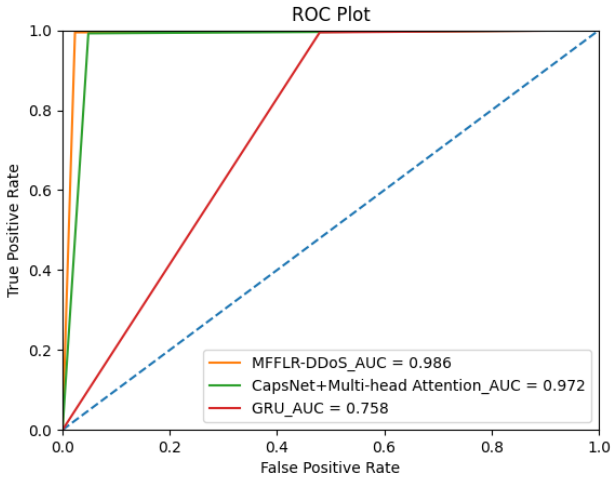


Figure 7. ROC comparison of different granularity models.

4.4.3. Performance comparison experiments of different models

To further evaluate the performance of the MFFLR-DDoS model, we compare our method with the state-of-the-art methods on the SlowDDoS and ISCX-2016-SlowDoS datasets, respectively. Table 6 shows the performance indicators of different methods on the SlowDDoS dataset, while Table 7 shows the performance indicators of different methods on the ISCX-2016-SlowDoS dataset.

Table 6. Performance metrics of different algorithms on SlowDDoS dataset.

Method	DR	FAR	F1-score
GMM-AE[22]	87.57%	22.51%	85.43%
CNN-Bi-LSTM[35]	96.49%	6.09%	95.95%
CNN-Bi-GRU[15]	96.36%	4.77%	96.36%
CapsNet-LSTM[23]	98.51%	8.06%	96.27%
MGREL[21]	98.79%	10.06%	95.70%
Our method	99.76%	3.02%	98.51%

Table 6 shows that the MFFLR-DDoS model has the highest detection rate on the SlowDDoS dataset, which is 1.25% and 0.97% higher than the CapsNet-LSTM model and the MGREL model, respectively. The false positive rate is the lowest, which is 3.07% and 1.75% lower than the CNN-Bi-LSTM model and the CNN-Bi-GRU model. The F1-score reaches 98.51%, which has the best overall classification performance among all models and is 2.15% and 2.24% higher than the CNN-Bi-GRU model and CapsNet-LSTM model, respectively. It shows that the MFFLR-DDoS model improves the ability to identify encrypted LR-DDoS attacks based on the effect of packet granularity and session granularity.

Table 7. Performance metrics of different algorithms on ISCX-2016-SlowDoS dataset.

Method	DR	FAR	F1-score
GMM-AE [22]	77.27%	9.59%	82.89%
CNN-Bi-LSTM [35]	72.89%	7.23%	83.99%
CNN-Bi-GRU [15]	77.91%	3.41%	87.43%
CapsNet-LSTM [23]	75.01%	8.13%	85.25%
MGREL [21]	92.64%	9.55%	91.83%
Our method	92.68%	3.47%	94.58%

As can be seen from Table 7, the MFFLR-DDoS model has the highest detection rate on the ISX-2016-SlowDoS dataset, and the detection rate reaches 92.68%, which is 0.04% higher than the MGREL model. The F1-score is also the highest among all models, which is 2.75% higher than that of the MGREL model. Moreover, the false positive rate is second to the CNN-Bi-GRU model, which is 0.06% higher than the latter. In general, MFFLR-DDoS has the best performance compared with other models. Our further analysis leads to the following conclusions:

1) The first four models deal with network traffic as a whole, and the classification effect is not good. The latter two models deal with network traffic in layers, and the detection rate and F1-score are both higher than 90%, which indicates that the hierarchical network model makes full use of the "session-packet" structural characteristics of network traffic, which is more conducive to anomaly classification.
2) The MFFLR-DDoS model combines CapsNet and GRU to mine the spatiotemporal features of traffic, which has a higher detection rate than the CNN-Bi-GRU model, indicating that the Capsule network has a better ability to extract spatial features than CNN.
3) MFFLR-DDoS and MGREL both introduce an attention mechanism; thus, they can capture

more important traffic characteristics than the non-attention model. Additionally, the multi-head attention mechanism can capture more dimensional key features than the single-head attention mechanism.

4.4.4. Defense mitigation

The MFFLR-DDoS detection system is added to the RYU controller to implement the above detection and defense mechanism. Figure 8 shows the throughput variation trend of the victim server h9 during an LR-DDoS attack. Before 28 s, host h9 processes the network packets sent by each real customer. When the LR-DDoS attack starts, at about 29 s, the throughput of h9 drops sharply. Because the connection thread of host h9 is occupied by the attacker, the legitimate client cannot request any service from the server, resulting in a rapid decrease in the rate at which the server processes packets. After about 6 s (the detection time includes data collection and preprocessing, feature extraction, and deep learning classification), the system detects the attack and initiates the mitigation module to locate the attack port and issue flow rules, and the h9 throughput starts to recover at about 39 s. It shows that the system realizes the defense purpose and can serve the real client normally, thus recovering the initial packet rate.

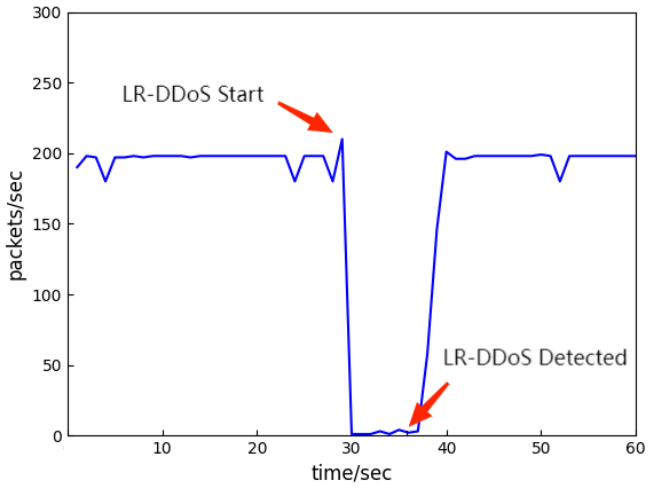


Figure 8. Effect of LR-DDoS attack mitigation.

5. Conclusions and future work

To effectively identify encrypted LR-DDoS attacks in SDN, this paper detects encrypted LR-DDoS attacks from a multi-granularity perspective. Initially, the traffic is divided based on the session granularity, and the spatial detail features of the session flow are extracted through the CapsNet model. Then, the weights are calculated by Multi-head Attention to extract the key session flow features. Then, the traffic is divided based on the packet granularity, and the timing features of the data packet are extracted by the GRU model. Finally, the features under different granularities are fused and the samples are classified, which can effectively improve the detection effect of encrypted malicious traffic. In addition, based on the idea that the SDN controller has centralized control, this paper proposes a real-time online encryption malicious traffic mitigation method, which can effectively alleviate the harm caused by attackers.

However, there are some shortcomings in this paper. In the future, we will focus on the

following three aspects: (1) More granular inputs are used to model encrypted traffic, improve the representation learning ability, and obtain a more comprehensive representation of encrypted malicious traffic. (2) Build a more reasonable data set. The data set should be collected in the real network environment, and the samples should be sufficient and representative. (3) An adaptive balanced training method is proposed to solve the problem of data imbalance.

Use of AI tools declaration

The authors declare that they have not used Artificial Intelligence (AI)tools in the creation of this article.

Acknowledgments

We thank the anonymous reviewers for their constructive comments, whose suggestions helped to improve the quality of our paper.

Conflict of interest

The authors declare no conflicts of interest in this paper.

References

1. Y. Zhang, L. Cui, W. Wang, Y. Zhang, A survey on software defined networking with multiple controllers, *J. Netw. Comput. Appl.*, **103** (2018), 101–118. https://doi.org/10.1016/j.jnca.2017.11.015
2. A. Dhanapal, P. Nithyanandam, The slow HTTP DDOS attacks: Detection, mitigation and prevention in the cloud environment, *Scalable Comput-Prac.*, **20** (2019), 669–685. https://doi.org/10.12694/scpe.v20i4.1569
3. M. Assis, L. Carvalho, J. Lloret, M. J. Proenca, A GRU deep learning system against in software defined network, *J. Netw. Comput. Appl.*, **177** (2021), 102942. https://doi.org/10.1016/j.jnca.2020.102942
4. S. Sabour, N. Frosst, G. Hinton, Dynamic routing between capsules, in *31st Annual Conference on Neural Information Processing Systems (NIPS)*, (2017), 3856–3866.
5. P. Kumar, R. Kumar, A. Kumar, A. Franklin, S. Garg, S. Singh, Blockchain and deep learning for secure communication in digital twin empowered industrial IOT network, *IEEE T. Netw. Sci. Eng.*, **10** (2023), 2802–2813. https://doi.org/10.1109/TNSE.2022.3191601
6. Y. Liu, T. Zhi, M. Shen, L. Wang, Y. K. Li, M. Wan, Software-defined DDoS detection with information entropy analysis and optimized deep learning, *Future Gene. Comput. Syst.*, **129** (2022), 99–114. https://doi.org/10.1016/j.future.2021.11.009
7. J. Bhayo, R. Jafaq, A. Ahmed, S. Hameed, S. Shah, A time-efficient approach toward DDoS attack detection in IoT network using SDN, *IEEE Int. Things J.*, **9** (2022), 3612–3630. https://doi.org/10.1109/JIOT.2021.3098029
8. Y. Cao, H. Jiang, Y. Deng, J. Wu, P. Zhou, W. Luo, Detecting and mitigating DDoS attacks in SDN using spatial-temporal graph convolutional network, *IEEE Trans. Depend. Sec. Cpmput.*, **19** (2022), 3855-3972. https://doi.org/10.1109/TDSC.2021.3108782

9. L. Zhang, J. Wang, A hybrid method of entropy and SSAE-SVM based DDoS detection and mitigation mechanism in SDN, *Comput. Secur.*, **115** (2022), 102604. https://doi.org/10.1016/j.cose.2022.102604
10. J. Wang, Y. Liu, H. Feng, IFACNN: Efficient DDoS attack detection based on improved firefly algorithm to optimize convolutional neural networks, *Math. Biosci. Eng.*, **19** (2022), 1280–1303. https://doi.org/10.3934/mbe.2022059
11. P. Chauhan, M. Atulkar, An efficient centralized DDoS attack detection approach for Software Defined Internet of Things, *J. Supercomput.*, **79** (2023), 10386–10422. https://doi.org/10.1007/s11227-023-05072-y
12. B. Gogoi, T. Ahmed, HTTP low and slow DoS attack detection using LSTM based deep learning, in *IEEE 19th India Council International Conference (INDICON)*, (2022), 1–6. https://doi.org/10.1109/INDICON56171.2022.10039772
13. B. Nugraha, R. Murthy, Deep learning-based slow DDoS attack detection in SDN-based networks, in *2020 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, (2020), 51–56. https://doi.org/10.1109/NFV-SDN50289.2020.9289894
14. N. Muraleedharan, B. Janet, A deep learning based HTTP slow DoS classification approach using flow data, *ICT Express*, **7** (2021), 210–214. https://doi.org/10.1016/j.icte.2020.08.005
15. C. Xu, J. Shen, X. Du. Low-rate DoS attack detection method based on hybrid deep neural networks, *J. Inf . Secur. Appl.*, **60**(2021),102879. https://doi.org/10.1016/j.jisa.2021.102879
16. Y. Chen, M. Zhang, F. Xu, Slow HTTP DoS attack detection method based on one-dimensional convolutional neural network, *J. Comput. Appl.*, **40** (2020), 2973–2979. https://doi.org/10.1109/MCG.2020.2973109
17. Y. Wang, R. Ye, Credibility-based countermeasure against slow HTTP DoS attacks by using SDN, in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, (2021), 890–895. https://doi.org/10.1109/CCWC51732.2021.9375911
18. N. Yungaicela-Naula, C. Vargas-Rosales, J. Perez, D. Carrera, A flexible SDN-based framework for slow-rate DDoS attack mitigation by using deep reinforcement learning, *J. Netw. Comput. Appl.*, **205** (2022), 103444. https://doi.org/10.1016/j.jnca.2022.103444
19. H. Li, S. Zhang, H. Song. W. Wang, Robust malicious encrypted traffic detection based with multiple features, *J. Cyber Secur.*, **6** (2021), 129–142.
20. A. Ferriyan, A. H. Thamrin, K. Takeda, J. Murai, Encrypted malicious traffic detection based on Word2Vec, *Electronics*, **11** (2022), 679–684. https://doi.org/10.3390/electronics11050679
21. Y. Gu, H. Xu, X. Zhang, Multi-granularity representation learning for encrypted malicious traffic detection, *Chin. J. Comput.*, **46** (2023), 1888–1899.
22. N. Garcia, T. Alcaniz, A. Gonzalez-vidal, J. B. Bernabe, D. Rivera, A. Skarmeta, Distributed real-time SlowDoS attacks detection over encrypted traffic using artificial intelligence, *J. Netw. Comput. Appl.*, **173** (2021), 102871. https://doi.org/10.1016/j.jnca.2020.102871
23. J. Tang, L. Yang, S. Liu, Caps-LSTM: A novel hierarchical encrypted VPN network traffic indentification using CapsNet and LSTM, in *3th International Conference on Science of Cyber Security (SciSec)*, (2021), 139–153. https://doi.org/10.1007/978-3-030-89137-4_10
24. M. Lotfollahi, M. Siavoshani, R. Zade, M. Saberian, Deep packet: A novel approach for encrypted traffic classification using deep learning, *Soft Comput.*, **24** (2020), 1999–2012. https://doi.org/10.1007/s00500-019-04030-2
25. S. Cui, J. Liu, C. Dong, Z. Lu, D. Du, Only Header: A reliable encrypted traffic classification framework without privacy risk, *Soft Comput.*, **26** (2022), 13391–13403. https://doi.org/10.1007/s00500-022-07450-9

26. Z. Zou, J. Ge, H. Zheng, Y. Wu, C. Han, Z. Yao, Encrypted traffic classification with a convolutional long short-term memory neural network, in *20th IEEE International Conference on High Performance Computing and Communications (HPCC)*, (2018), 329–334.
27. H. Yan, J. Wang, P. Zhang, Capsule network assisted IoT traffic classification mechanism for smart cities, *IEEE Int. Things J.*, **6** (2019), 7515–7525. https://doi.org/10.1109/JIOT.2019.2901348
28. Y. Zeng, H. Gu, W. Wei, Y. Guo, Deep-full-range: A deep learning based network encrypted traffic classification and intrusion detection framework, *IEEE Access*, **6** (2019), 45182–45190. https://doi.org/10.1109/ACCESS.2019.2908225
29. SplitCap tool. Available from: https://www.netresec.com/index.ashx?Page=SplitCap.
30. B. Nunes, M. Mendoca, X. Nguyen, K. Obraczka, T. Turletti, A survey of software-defined networking: Past, present, and future of programmable networks, *IEEE Commun. Surv. Tut.*, **16** (2014), 1617–1634. https://doi.org/10.1109/SURV.2014.012214.00180
31. R. De Oliveira, A. Shinoda, C. Schweitzer, L. Prete, Using mininet for emulation and prototyping software defined networks, in *2014 IEEE Colombian Conference on Communications and Computing (COLCOM)*, (2014), 1–6. https://doi.org/10.1109/ColComCon.2014.6860404
32. Slowhttptest Tool Source Code. Available from: https://github.com/shekyan/slowhttptest/.
33. D-ITG Tool User Guide. Available from: http://traffic.comics.unina.it/software/ITG/manual/
34. H. Jazi, H. Gonzalez, N. Stakhanova, A. Ghorbani, Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling, *Comput. Netw.*, **121** (2017), 25–36. https://doi.org/10.1016/j.comnet.2017.03.018
35. Z. Liu, J. Yu, B. Yan, G. Wang, A deep 1-D CNN and bidirectional LSTM ensemble model with arbitration mechanism for LDDoS attack detection, *IEEE Trans. Emerg. Top. Comput. Intell.*, **6** (2022), 1396–1410. https://doi.org/10.1109/TETCI.2022.3170515

AIMS Press