AcLGB: A Lightweight DDoS Attack Detection Method

Fantao Zeng[1,2], Jieren Cheng[1,2(✉)], Zhuyun Cao[1], Yue Yang[1,2], and Victor S. Sheng[3]

[1] School of Computer Science and Technology, Hainan University, Haikou 570228, China
cjr22@163.com

[2] Hainan Blockchain Technology Engineering Research Center, Haikou 570228, China

[3] Department of Computer Science, Texas Tech University, TX 79409, USA

Abstract. With the development of Internet technology, distributed denial of service(DDoS) attack has always been a hot and difficult point in network security.Protecting network infrastructure and information security is also becoming more and more important.However, cyber security is an arms race, as attacks develop and network traffic surges, intelligent solutions face the challenge of detecting sensitive changes in traffic characteristics.In this paper, we propose a lightweight Adaptive Clustering-based LightGBM(AcLGB) detection method.This is a new DDoS traffic classification method and an effective lightweight detection method.We introduce a new clustering technique to learn the clustering centers that can be used to extend the characteristics of a given dataset.It solves the challenge of difficult detection when traffic characteristics change sensitively.The model separates the samples of different categories in the best way, and outperforms the current detection method with 99.98% detection accuracy. In the CIC-DDoS2019 data set, the detection time of 802s is better than other detection methods.

Keywords: Network security · DDoS attack · LightGBM · Clustering

1 Background Introduction

1.1 Background

A distributed denial of service (DDoS) attack involves flooding a target server with traffic, rendering it inoperable. It is different from other attack types. The main purpose of DDoS attacks is not to steal private data, but to degrade the performance of the target server. DDoS attacks are distributed denial-of-service attacks. Multiple clients attack a target server at the same time, rapidly depleting the resources of the target server. Botnets of malware-infected client computers are also a form of DDoS attack.

X. Qiu et al. (Eds.): ISCC 2023, SIST 350, pp. 200–212, 2024.
https://doi.org/10.1007/978-981-99-7161-9_16

The most recent massive DDoS attack was a massive attack on GitHub in 2018, which lasted 20 min. GitHub has its own internal security mechanism that blocks attacks when attacked, and it was one of the largest DDoS attacks in the world. In 2022, Google Cloud Armor customers suffered a distributed denial of service (DDoS) attack based on the HTTPS protocol that reached 46 million requests (RPS) per second, the largest attack of its kind ever recorded. In just two minutes, the attack escalated from 100,000RPS to a record 46 million RPS, nearly 80 percent higher than the previous high, and Cloudflare eased an HTTPS DDoS of 26 million RPS in June. The attack began at 09:45 am Pacific time on June 1, initially targeting the victim's HTTP/S load balancer at a rate of 10,000RPS. Within eight minutes, the attack intensified to 100,000RPS, and Google's CloudArmorProtection was activated by generating alerts and signatures based on certain data extracted from traffic analysis. Two minutes later, the attack peaked at 46 million requests per second. Fortunately, the customer had already deployed Cloud Armor's recommendation rules, and the hack did not have the desired effect.From this, we can see that DDoS attacks are not far away from us, and let's look at the consequences of DDoS attacks.

First Half of 2022 Attack Statistics

Reflected Attack Destinations

NEXUSGUARD®

Top Ten Reflected Attack Destinations around the globe (1HY 2022)

	Percentage
Brazil	40.60%
South Korea	34.01%
China	6.03%
United States	5.99%
Ecuador	1.67%
United Kingdom	1.53%
Kazakhstan	0.83%
Russian Federation	0.73%
Canada	0.67%
Seychelles	0.63%
Others	7.30%

Fig. 1. Top ten reflected attack destinations around the globe(1HY 2022)

According to NexusGuard [1] statistics on DDoS attacks in 2022, the total number of attacks and average attack size increased by 75.60 percent and 55.97 percent, respectively, in the first half of 2022 compared to the second half of 2021. Compared to the second half of 2021, the maximum attack size was reduced by 66.82%, and the maximum attack size was 232.00 Gbps. Compared to the same month in five years, March had the fewest number of attacks, while June had the highest number of attacks, the highest number of attacks, and the highest number of attacks. While the number of attacks increased from April 2022 to June 2022, the number of attacks declined during the same period in 2021. Figure 1 shows this.

We summarize the data of DDoS attacks in the following sections.

a. Type of attack media: In the first half of 2022, UDP attack and HTTPS flood were the two attack types, contributing 39.58% and 15.94% respectively, while TCP ACK attack ranked the third with 6.48%.
b. Attacks by category: Volume (direct flood) attacks accounted for 67.93% of total attacks recorded in the first half of 2022, with HoH increasing 48.22% and down 15.06% year-on-year.
c. Protocol attacks: UDP and TCP attacks were the two main attack types in the first half of 2022, accounting for 61.27 and 30.57% respectively.
d. Attack duration: 69.27% of attacks lasted less than 90 min, and the remaining attacks lasted more than 90 min. 17.15% of the attacks lasted more than 1200 min.

From the above data, we can find that the target of DDoS attacks is not only to affect the target website, but also to affect the normal operation of services. The cost of DDoS attacks is relatively low, but large-scale DDoS attacks have a huge impact on services. Because attackers often change the nature of attacks, such sensitive characteristics make it difficult to detect DDoS attacks, and therefore difficult to detect and mitigate the impact of attacks.

1.2 Main Contribution

Despite rapid advances in AI-based DDoS detection methods [11], existing solutions are still very sensitive to small changes in the various characteristics of network traffic. Specifically, because these techniques learn from the characteristics of a single sample, training them on carefully designed features, inadvertently mislabeled samples, or small subsets of unbalanced data sets negatively affects their ability to generalize, thus making it very difficult to detect new DDoS attacks.

Main contribution: We introduce a new clustering technique to learn clustering centers that can be used to extend the characteristics of a given data set.Based on the clustering results, we use the normalization method of softmax to process the data set, which is convenient for lightGBM classifier to conduct classification training. The statistical features and clustering features are joined together, and then LightGBM algorithm is applied to classify the generated new data set.Finally, in order to prove the effectiveness of our solution, we evaluated the effectiveness of the AcLGB algorithm on the network traffic data set of CIC-DDoS2019 [2], and it reached the accuracy of 99.98979% in the detection accuracy. In terms of detection time, 802s is better than the conventional classification model.

2 Related Work

We briefly describe other test methods that are directly related to our test method and highlight their shortcomings.

2.1 DDoS Attack Detection Based On Deep Learning

Most deep learning-based detection methods attempt to match observed network flow with previously learned patterns. Despite increasing adoption rates, they produce unacceptably high false positive rates with relatively little improvement in detection performance. This greatly limits their applicability in real life. Autoencoders (AE) can learn potential representations of features and reduce their dimensions to minimize memory consumption [3–5], which drives their use for abnormal traffic detection. Tan et al. [6] applied convolutional neural networks (CNN) to learn the spatial representation of packets, and then used image classification methods to identify malware traffic. Wang et al. [7] combined CNN with long short-term memory (LSTM) structures to learn the spatial and temporal correlations between features. As effective as these techniques are, they completely ignore the time-based statistical characteristics that can be inferred from the semantic relationships in packets and packet payloads. Min et al. [8] used these ignored attributes and applied natural language processing techniques to process the packet payload. This improves detection performance, but it still has several important weaknesses, including ignoring data set imbalances and showing very high processing times when working with large data sets.

2.2 DDoS Attack Detection Based On Machine Learning

Machine learning refers to the analysis of large amounts of data by machines and the automatic learning of rules and patterns in the data, so as to achieve automatic decision-making and control. In the field of DDoS attack detection, many scholars carry out researches. Dong et al. [9] proposed the improved KNN (K-Neighbors), which mainly focuses on adding a weight to the predicted samples. The weight can make the samples that are closer to the predicted samples contribute more to the model, and the algorithm can perform better in some specific distributions. Due to the defects of KNN itself, its efficiency in processing large samples will be insufficient. Li et al. [10] proposed a method of feature dimension reduction, which extracted 19-dimensional features from high-dimensional network traffic and classified network traffic by combining clustering and support vector machine (SVM) algorithm. In the field of machine learning, feature selection is often decided by humans. Once the feature selection is insufficient, the training effect will be poor.

3 System Architecture

A lightweight AcLGB detection method based on LightGBM.

We propose a lightweight DDoS attack detection method based on LightGBM, which can maximize the detection efficiency of the model and reduce the detection time. At the same time, multi-core cluster balancing detection performance is introduced to ensure that the detection accuracy is not lost. It not only reduces the false alarm rate, but also improves the detection efficiency.

Feature extractor module: converts raw network packets into headers and statistical feature vectors.

Adaptive clustering module: low dimensional embedding of network flow features is constructed, and abstract attributes shared by a group of samples belonging to the same traffic type are calculated.

Classification module: statistical features and clustering features are connected together, and then LihgtGBM algorithm is applied to classify the generated new data set.

In the following sections, we describe in detail the specific operations of each module and its related role in detecting DDoS attacks.

3.1 Feature Extractor Module

By processing the data features of the original data set, AcLGB extracts more representative representation information from it to represent the data, which is embodied in: deleting null and 0 values; Remove useless features including "Flow ID", "Source IP", "Unnamed: 0", etc. By reducing useless features and reducing the training pressure of classifiers, the performance of clustering modules and classifiers can be effectively improved, and the training reasoning time of AcLGB can be greatly reduced.

3.2 Adaptive Clustering Module

Although the training speed of LightGBM is very fast and the memory usage is very small, it is sensitive to the interference effect of noisy data. We need to organize the data input effectively to reduce the impact of data on the classification module. We propose a new model, AcLGB, which is based on a clustering algorithm that generates clustering centers to be used as extensions of the input features to be clustered. Because our adaptive clustering method is designed to be end-to-end differentiable, the training is performed on small batches by which the network learns the low-dimensional representation of the input and computes the corresponding kernel center. This operation is performed online iteratively, with the probability that the last layer of the kernel network produces each sample among inputs belonging to all possible classes.

The clustering algorithm is to learn the similarity content of the data set to group similar samples together. Through this process, the data is effectively organized. By proposing multiple kernel clustering networks, each network may learn one of the required classification clusters, and we use encoders to reduce the original network traffic dimension to any required dimension. As shown in the Fig. 2, we can use the encoder to reduce the dimensionality of the input features to the desired feature dimension.

Let Kc be the set of features obtained from the cluster center of a given class C. For any two samples i, j $\in$ C.

$$\text{distance } (\chi_i \cup K_c, \chi_j \cup K_c) \leq \text{ distance } (\chi_i, \chi_j) \tag{1}$$

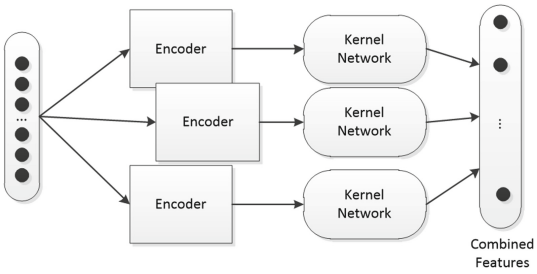


Fig. 2. Multiple kernel clustering process

Let i and j be n-dimensional real-valued vectors:

$$\chi_i = \{x_{i1}, x_{i2}, \ldots, x_{in}\} \in \mathbb{R}^n \quad \chi_j = \{x_{j1}, x_{j2}, \ldots, x_{jn}\} \in \mathbb{R}^n \tag{2}$$

When Xit and Xjt correspond to each feature of the sample data, t belongs to [0, n], assuming that xi and xj are both C-class data, it is clear that we can obtain the CTH cluster center Kc:

$$K_c = \{k_{c1}, k_{c2}, \ldots, k_{cm}\} \in \mathbb{R}^m \tag{3}$$

Finally we obtain the aggregated features:

$$\chi_i' = \{x_{i1}, x_{i2}, \ldots, x_{in}, k_{c1}, k_{c2}, \ldots, k_{cm}\} \in \mathbb{R}^{n+m} \tag{4}$$

$$\chi_j' = \{x_{j1}, x_{j2}, \ldots, x_{jn}, k_{c1}, k_{c2}, \ldots, k_{cm}\} \in \mathbb{R}^{n+m} \tag{5}$$

An intuitive way for us to calculate the clustering effect is to compare the similarity between the original and aggregated feature vectors, namely Q1 and Q2,

$$Q_1 = \text{distance}\,(\chi_i, \chi_j) \text{ and } Q_2 = \text{ distance } \left(\chi_i', \chi_j'\right)$$

$$Q_1 = \frac{1}{n} \sum_{\alpha=1}^{n} \left(x_{i\alpha} - x_{j\alpha}\right)^2 \tag{6}$$

$$Q_2 = \frac{1}{n+m} \left(\sum_{\alpha=1}^{n} \left(x_{i\alpha} - x_{j\alpha}\right)^2 + \sum_{\alpha=1}^{m} \left(k_{c\alpha} - k_{c\alpha}\right)^2 \right) \tag{7}$$

$$Q_2 = \frac{1}{n+m} \sum_{\alpha=1}^{n} \left(x_{i\alpha} - x_{j\alpha}\right)^2 \tag{8}$$

$$Q_1 - Q_2 = \frac{m}{n+m} \cdot Q_1 \Rightarrow Q_2 = \beta \cdot Q_1 \tag{9}$$

3.3 Classification Module

Finally, AcLGB uses classification module to process the combined extraction features and clustering center of each sample, and outputs the inferred traffic class to which the flow belongs. The training classification is carried out by LightGBM. 70% of the data set is taken as the training set and 30% is taken as the verification set to judge and classify the input data. LightGBM is a classification algorithm based on decision tree. First, it is a decision tree algorithm based on Histogram. It discretized continuous floating point feature values into k integers, constructs histogram with width of k, and performs statistics in the histogram according to discretized values as indexes. Then, according to the discrete value of the histogram, the optimal segmentation point is found by traversing. LightGBM conducts difference acceleration based on Histogram algorithm. Secondly, LightGBM adopts a growth strategy based on the leaf-wise algorithm with depth limitation. This strategy finds the Leaf with the largest splitting gain from all current leaves each time, and then splits, and so on. On this basis, the maximum depth limit is added to prevent overfitting on the basis of ensuring high efficiency.

Compared with traditional XGBoost, LightGBM has the advantages of faster speed and smaller memory, so we choose this classification algorithm as the design of our classification module.

3.4 AcLGB Detection Procedure

Lightgbm-based lightweight attack detection method mainly uses LightGBM's lightweight detection model to improve the overall detection speed and ensure the memory overhead, and reduces the impact of noise data in the data set on the training process through multi-core clustering to ensure the overall robustness. As shown in the Fig. 3, the specific process is as follows:

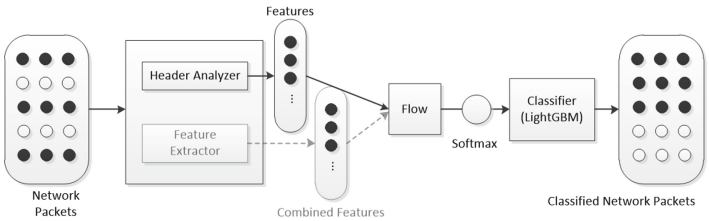


Fig. 3. AcLGB detection procedure

Step 1 AcLGB processes very large volumes of traffic by processing the raw network packet stream into a feature-based bidirectional stream representation.

Step 2 AcLGB is based on the clustering method. We extract a series of features from the original data packet and extend these features using the learned clustering center.

Step 3 AcLGB uses softmax normalization method to process the input training data.

Step 4 AcLGB uses LightGBM classifier to detect DDoS attacks.

4 Specific Experiments

In this section, we introduce the specific process of the experiment.

4.1 Dataset

We evaluate the performance of our method on CIC-DDoS2019 dataset. CIC-DDoS2019 contains benign, up-to-date common DDoS attacks that resemble real real-world data. It also includes the results of network traffic analysis. Using CICFlowMeter-V3, streams are marked according to timestamp, source and destination ip, source and destination port, protocol, and attack. It is collected by the Canadian Network Security Laboratory on January 12, 2019 and March 11, 2019. The total data set is 28.9GB. The data set contains a variety of DDoS attack classification labels. Make an experimental comparison. Finally, we split the preprocessed dataset into a training set and a test set with a ratio of 70 and 30%.

4.2 Preprocessing

Through Fig. 4, we can see that there are many data features that are useless for our model training, so we remove null and zero values; Remove useless features, including "Flow ID", "Source IP", "Unnamed: 0", etc. By reducing useless features and reducing the training pressure of the classifier, the performance of the clustering module and classifier can be effectively improved, and we greatly reduce the training inference time of AcLGB.

4.3 Experimental Parameter Selection

We implemented AcLGB in Python 3.7 using PyTorch [16] and LightGBM libraries. For the AcLGB algorithm, we adopted a set of fully connected NN encoders with 3 hidden layers containing 500, 200 and 50 neurons, respectively. The number of neurons in the output layer is equal to the required dimension of the kernel, which we set to 10 in our experiments with e^{-4} as the training learning rate. We use the classification algorithm of LightGBM as our classifier, and the parameters of LightGBM are shown in Table 1.

Next we evaluate the results of the overall DDoS attack detection, where AcLGB extrapolates meaningful low-dimensional representations from headers and statistical features extracted from raw network traffic data. Using these automatically learned features, we determine different cluster centers and use them to expand the title and statistical properties. With these additional features, we expect our classifier to be more accurate and to easily distinguish even the most

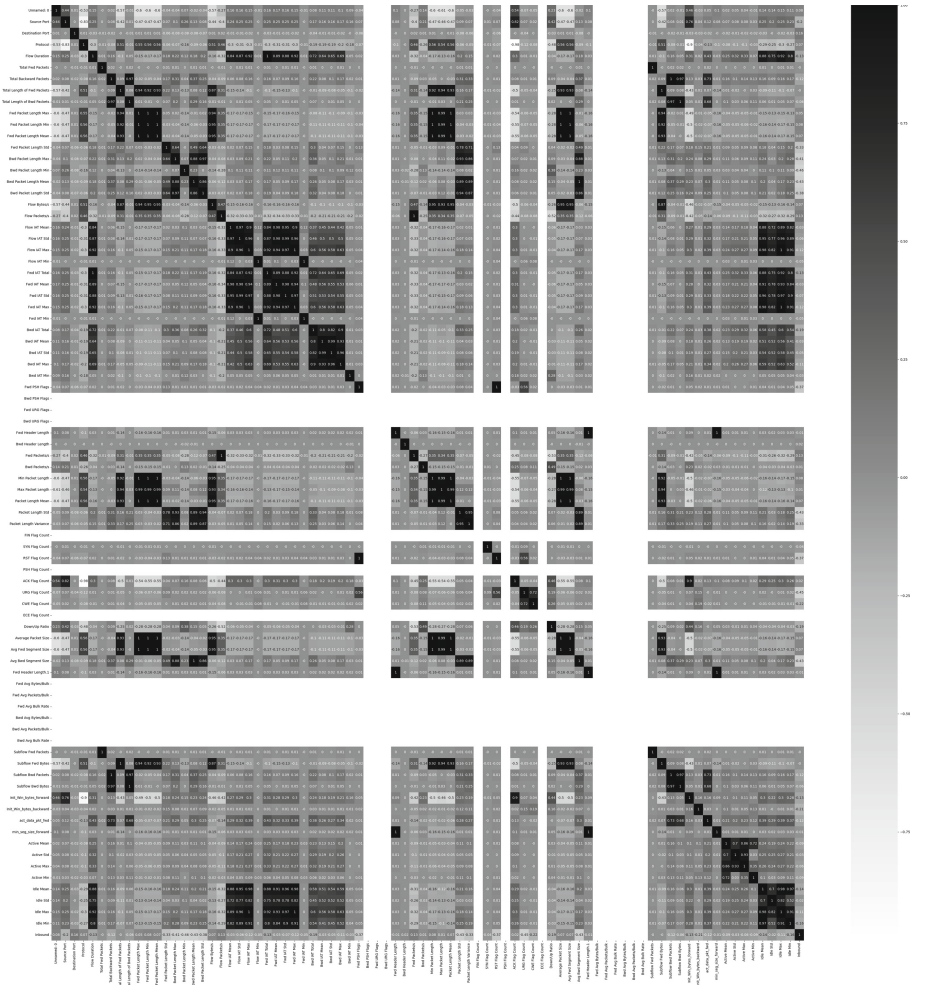
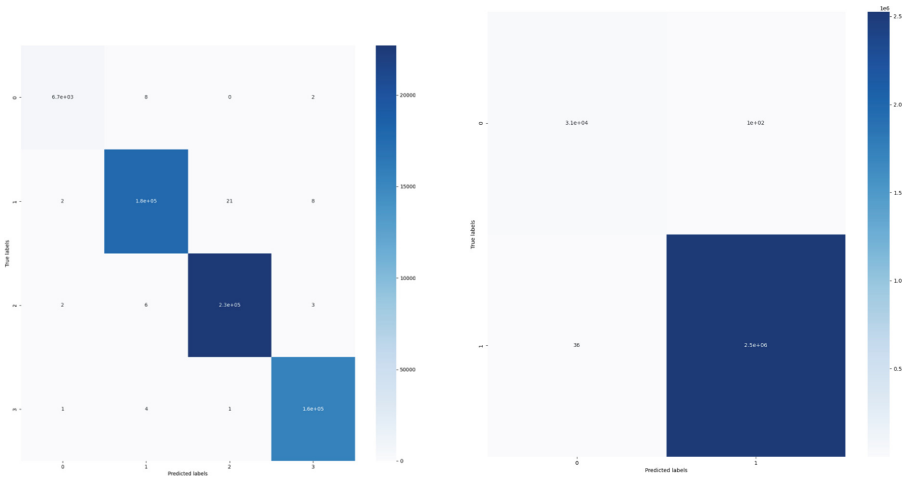


Fig. 4. Feature correlation graph

Table 1. Hyperparameter selection of LightGBM

Parameters	Default values	Meaning
boosting_type	gbdt	Set the promotion type
num_leaves	31	Number of leaf nodes
learning_rate	0.1	Learning rate
feature_fraction	0.9	Feature selection ratio
begging_fraction	0.7	Sample sampling ratio
begging_freq	5	Iterative execution cycle

similar patterns. To verify our hypothesis, we perform multi-label classification on the above CIC-DDoS2019 real dataset, as shown in Fig. 5a. Therein we distinguish between benign and malicious traffic and identify the type of each traffic separately. And do the binary classification experiment without using multiple kernel clustering to process the data, as shown in Fig. 5b.



(a) 4 Classification results (b) 2 Classification results

Fig. 5. Confusion matrix results

Table 2. Confusion matrix in classification task

	Actual positive	Actual negative
Predicted positive	TP	FP
Predicted negative	FN	TN

4.4 Evaluation Metrics

To measure the performance of our AcLGB, we use the held out testing set to compute confusion matrices, based on which we calculate the number of True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) inferences. The details are given in Table 2. With these, we derive a number of

metrics that allow us to assess the quality of the classification results of AcLGB and those produced by the benchmarks considered, namely:

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN} \tag{10}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{11}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{12}$$

$$\text{FAR} = \frac{FP}{FP + TN} \tag{13}$$

$$\mathbf{F_1score} = 2 \cdot \frac{\text{Precision*Recall}}{\text{Precision+Recall}} \tag{14}$$

In the CIC-DDoS2019 real dataset, we also calculated the accuracy, precision, recall, F1 score of AcLGB, which is similar to a two-stage classification process, where the first stage corresponds to classifying DDoS traffic by clustering. The second stage corrects misclassified samples by a further LightGBM classifier. Finally, we compare the experimental performance of different methods on CIC-DDoS2019 dataset in Table 3, We can find that our method has a detection accuracy of 99.98%, a Recall value of 99.71% and a F1 score of 99.55%, which is much higher than other detection methods.

Table 3. Performance comparison of different methods on CIC-DDoS2019 dataset

Thesis	Methods	Accuracy	Precision	Recall	F1 score	Training time
De Assis MOV [12]	CNN+LSTM	96.34	95.71	95.49	95.60	1077
Javaid A [13]	Regression	95.59	95.41	95.95	95.53	1245
Sadaf K [14]	Isolasion Forest	91.49	90.28	90.74	90.51	1377
Wei Y [15]	AE+MLP	97.76	97.74	97.63	97.68	1127
AE-XGBoost	AE+XGBoost	98.92	98.96	98.94	98.95	1745
Our	AcLGB	99.98	99.38	99.71	99.55	802

5 Conclusion and Prospect

Based on LightGBM, this paper proposes a lightweight DDoS attack detection method AcLGB, which can maximize the detection efficiency of the model and reduce the detection time. At the same time, multi-core clustering is introduced to balance the detection performance and ensure that the detection accuracy is not lost. It not only reduces the false positive rate, but also improves the detection efficiency. We verify the effectiveness of our method through experiments on

CIC-DDoS2019 dataset, which can be deployed on devices with limited computing resources. At present, the abnormal flow of this data set is much higher than the normal flow. In the future, the data set with more balanced distribution can be found to verify the method.

This work was supported by National Natural Science Foundation of China (NSFC) (Grant No. 62162022, 62162024), the Key Research and Development Program of Hainan Province (Grant No. ZDYF2020040, ZDYF2021GXJS003),the Major science and technology project of Hainan Province (Grant No. ZDKJ2020012), Hainan Provincial Natural Science Foundation of China (Grant No. 620MS021, 621QN211), Science and Technology Development Center of the Ministry of Education Industry-university-Research Innovation Fund(2021JQR017), Innovative research project for Graduate students in Hainan Province (Grant No. Qhyb2022-93). Supported by the Key Laboratory of PK System Technologies Research of Hainan.

References

1. "NexusGuard" [online] Available: https://www.netscout.com/
2. "CIC-DDoS2019" [online] Available: https://www.unb.ca/cic/datasets/ddos-2019.html
3. Yu, Y., Long, J., Cai, Z.: Session-based network intrusion detection using a deep learning architecture. In: International Conference on Modeling Decisions for Artificial Intelligence, pp. 144–155 (2017)
4. Yu, Y., Long, J., Cai, Z.: Network intrusion detection through stacking dilated convolutional autoencoders. In: Security and Communication Networks, pp. 1–10 (2017)
5. Yousefi-Azar, M., Varadharajan, V., Hamey, L., Tupakula, U.: Autoencoder-based feature learning for cyber security applications. In: IEEE International Joint Conference on Neural Networks, pp. 3854–3861 (2017)
6. Tan, Z., Jamdagni, A., He, X., Nanda, P., Liu, R., Hu, J.: Detection of denial-of-service attacks based on computer vision techniques. IEEE Trans. Comput. **64**(9), 2519–2533 (2014)
7. Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., et al.: HAST-IDS: Learning Hierarchical Spatial-Temporal Features using Deep Neural Networks to Improve Intrusion Detection (2017)
8. Min, E., Long, J., Liu, Q., Cui, J., Chen, W.: TR-IDS: anomaly-based intrusion detection through text-convolutional neural network and random forest. In: Security and Communication Networks, pp. 1–9 (2018)
9. Dong, S., Sarem, M.: DDoS attack detection method based on improved KNN with the degree of DDoS attack in software-defined networks. **8**, 5039–5048 (2018)
10. Li, Y., Xia, J., Zhang, S., et al.: An efficient intrusion detection system based on support vector machines and gradually feature removal method. Expert Syst. Appl. **39**(1), 424–430 (2012)
11. Cui, J., Zhang, Y., Cai, Z., Liu, A., Li, Y.: Secure-display path for security-sensitive applications on mobile. Comput. Mater. Continua **55**(1), 17–35 (2018)
12. de Assis, M.V.O., Carvalho, L.F., Rodrigues, J.J.P.C., et al.: Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. Comput. Electr. Eng. **86**, 106738 (2020)

13. Javaid, A., Niyaz, Q., Sun, W., et al.: A deep learning approach for network intrusion detection system. Eai Endorsed Trans. Secur. Saf. **3**(9), 2 (2016)
14. Sadaf, K., Sultana, J.: Intrusion detection based on autoencoder and isolation Forest in fog computing. **8**, 167059–167068 (2020)
15. Wei, Y., Jang-Jaccard, J., Sabrina, F., et al.: AE-MLP: a hybrid deep learning approach for DDoS detection and classification. IEEE **9**, 146810–146821 (2021)
16. "PyTorch", [online] Available: https://pytorch.org/