A Programmable Switch Based Detection of DDoS Attacks Intelligent Network Element Realization

Jingfu Yan, Weilin Wang, and Huachun Zhou*

Beijing Jiaotong University, Beijing, China
{22110030, 21111026, hchzhou}@bjtu.edu.cn

Abstract

Combining AI applications with web security devices improves the network's ability to detect and respond to DDoS attacks. However, the current AI application is a kind of plug-in AI that cannot improve the performance of the 6G network itself. In addition, AI applications need to consume the transmission bandwidth when analyzing traffic ,as well as the computation and storage resources of security devices, that cannot meet the real-time DDoS attack detection and processing requirements. In this paper, we propose an intelligent network element built by combining programmable switches and AI algorithms to realize network intelligence at the network layer by analyzing the packet header information of 6G traffic to classify packets. The packets of fiv e types of DDoS, separately, AL DDoS, LDDoS, Botnet, Reflection DDoS, and TL DDoS, are analyzed by sinking the ML model to the network layer. Finally, by comparing the classification performance of DT model, RF model and S-RF model proposed in this paper, deployed in INE, we concluded that the S-RF algorithm performs best in the traffic detection.

Keywords: Programmable Switches, DDoS Traffic, Machine Learning

1 Introduction

Cloudflare's DDoS (Distributed Denial of Service) Trends Report for Q1 2023 shows that DDoS attacks above 10Gbps continue to show growth globally [1].DDoS attacks are designed to consume network bandwidth or system resources, causing the network or system to become overloaded and stop providing normal network services, which can be a serious threat to Internet security. With the introduction of AI(Artificial Intelligence) algorithms into security networks in the form of applications, the detection and generalization capabilities of security networks for DDoS attacks have been improved [2] . However, this is a kind of plug-in referencing for AI algorithms that cannot enhance the intelligence of the network architecture. The plug-in AI application cannot detect DDoS attacks in the network in real time or process different types of traffic in real time.

INE(Intelligent Network Element) is obtained by deconstructing AI models and embedding them into network elements located in the network layer, which have certain perception, reasoning, and auxiliary model training capabilities [3]. 6G networks place new demands on the network architecture, and unlike the way AI is applied in 5G networks, 6G networks need to deeply integrate the network architecture with AI in order to meet the objectives of improving network performance through AI and enhancing AI services through intelligent networks [4] . The network composed of INE can form distributed AI at the edge of the network, maximizing AI performance and effectively solving problems such as data detection, computing resources,

The 7th International Conference on Mobile Internet Security (MobiSec'23), Dec. 19-21, 2023, Okinawa, JAPAN, Article No. 48

*Corresponding author: Beijing Jiaotong University, No.3 Shangyuancun, Haidian District, Beijing 100044, China, Email: hchzhou@bjtu.edu.cn Tel: +86-137-1816-8186

and storage resources [5] .The network can be used as a network of INE. For this reason, by building an intelligent network architecture composed of INE, the detection capability of the network itself can be improved and network resources can be saved.

This paper combines programmable switch technology with AI technology to design an intelligent network element that can detect DDoS attacks by building ML (Machine Learning) algorithms inside the programmable switch. Using BMv2 as data plane and P4Runtime as control plane, AI is sunk into the network layer to detect multiple types of DDoS attacks. After that, in order to obtain the best ML model parameters, the packet headers obtained from the programmable switches are used as the features of the training inputs, as well as, the paper all machine learning models to be deployed are trained offline. Finally, this paper performs classification tests on INE deployed with different AI and compares the test results.

The rest of the paper proceeds as follows. Section 2 briefly summarizes the related work on DDoS detection. Section 3 explains the process design and implementation methodology of this research paper. Section 4 describes the evaluation scheme and comparison of results of our approach. Section 5 summarizes the evaluation results and suggests research directions for future work. Section **??** describes the program support for our work.

2 Related Work

DDoS attacks can be categorized into LDDoS (Low-rate DDoS) and capacity depletion DDoS based on traffic characteristics and resource consumption. Capacity depletion DDoS include AL DDoS (Application Layer DDoS), Botnet, DRDoS (Distributed Reflection DoS), TL DDoS (Transport Layer DDoS), and other attacks[6]. The main task of DDoS attack detection is to discover DDoS attacks in the network, generate warnings, and trigger an attack response mechanism [7].

Existing DDoS detection methods mitigate to some extent problems such as insufficient bandwidth resources caused by DDoS attacks in the network. However, the defense against DDoS attacks still faces dilemmas such as external AI, inability to process traffic in a timely manner, and lack of immediate feedback. DDoS attack detection methods can be divided into three types: statistics-based DDoS attack detection, AI algorithm-based DDoS attack detection, and intelligent network element-based DDoS attack detection.

Statistical-based DDoS attack detection methods are suitable for detecting high-capacity attacks, and statistical criteria such as correlation, entropy, and covariance are commonly used to analyze network traffic and detect anomalous patterns [8].Wang Z et al. detected high-capacity network attacks by analyzing the correlation of IP addresses [9]. Tsibdjou L D et al. detected the normalized Shannon entropy of Internet Protocol addresses to determine whether the whether the traffic activity is suspicious or not [10].Lin H et al. designed a covariance based RM matrix to detect application layer DDoS attacks by recording access traces in RM [11]. Statistical-based DDoS attack detection methods are good at suppressing high-capacity attacks, but they are prone to misclassify normal traffic as malicious traffic when there is a traffic flood, and they are very ineffective at detecting the more insidious DDoS attacks.

AI-based DDoS detection can be categorized into ML-based DDoS detection and deep learning-based DDoS detection based on the different AI algorithms, and the most obvious difference between the two is whether or not they use convolutional neural networks. Sahoo K S et al. used a genetic algorithm to optimize the parameters of the SVM algorithm and deployed the SVM model in a controller to develop security rules for DDoS attack detection [12].Alsirhani A et al. analyzed the effectiveness of four types of ML, DT(Decision Tree)(Gini), DT(Entropy), RF(Random Forest), and Simple Bayes, in classifying DDoS attacks in a static

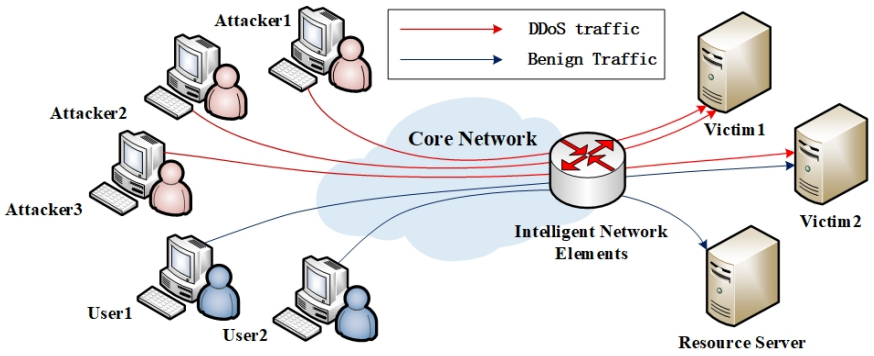

Figure 1: Intelligent network element deployment

Spark system [13] .Zainudin A et al. used CNN+LSTM to analyze the traffic in CICDDoS2019 dataset[14] . It can be seen from the above et al. study that most of the existing detection methods use out-of-the-box AI, which leads to their conclusions being obtained based on the analysis of model performance in a static environment. Although some AI models are deployed in controllers in some distributed SDN network architectures, the consumption of transmitting traffic information in the network itself is also extremely high.

INE based DDoS attack detection uses a combination of programmable switches and AI technology, which retains the advantages of AI technology in the detection field and achieves real-time detection. Depending on the AI algorithm used, it can be categorized into sketch based DDoS detection, machine learning based DDoS and convolutional neural network based DDoS detection. Detection Ding D et al. proposed a BACON Sketch approach, which uses a hash algorithm to convert traffic features into a two-dimensional sketch, and achieves DDoS detection by counting the changes in the sketch, but the approach requires switch computational and storage resources for generating and storing sketches [15].He W et al. designed a Bayesian algorithm based DDoS detection framework, but it still needs to collect data in the network to determine the traffic type centrally [16].Cao Y et al. proposed a DDoS detection model based on spatio-temporal graph convolutional network, which uses the extracted spatio-temporal features to find the attack path. The method requires the collection of a large amount of traffic information and uses an entropy-based static judgment approach in intercepting DDoS attacks, and the graph neural network is only used to trace back the attack [17].

It can be seen that INEs are more suitable for dynamic network environments and real-time changing traffic characteristics than statistical and AI-based DDoS attack detection methods. Sketch-based and convolutional neural network-based DDoS detection methods require high storage and computation capacity of the switch itself, while ML-based detection methods can take up less storage space and computation resources to realize the detection function of INEs.

3 Research Design and Realization

In this section, we first describe the deployment scenario of INEs. We then detail the offline training process and the S-RF algorithm. Finally, we demonstrate the structure of INE utilizing BMv2 to deploy S-RF.

3.1 INE Deployment

As Figure 1 shown, we target the ISP (Intrusion Prevention Systems) network and deploy an intelligent NE with DDoS detection capability at the edge ingress of the core network near the resource server, so that at least one intelligent NE in the network composed of intelligent NEs can receive traffic to the resource server. In other words, at least one intelligent NE can detect traffic from the source host that sent the traffic. Once a DDoS attack is detected, packets are classified or dropped.

3.2 Offline Training

The purpose of offline training is to obtain the optimal parameters of the ML model so that the model deployed in the programmable switch has the best classification performance. The process of training the model includes four processes: feature extraction, feature preprocessing, feature selection and model training.

The raw packets need to be filtered before feature extraction, and since only some kinds of traffic are used in this paper, the scapy library in Python is utilized to filter by IP address.

Feature extraction requires extracting packet headers consistent with programmable switches as features, for this purpose this paper uses the scapy library in the python environment to process the packets. Using this algorithm we extracted a total of six categories of headers including Ethernet, IP, UDP, ARP, TCP, IPv6, and ICMP, but the header features of the four categories of ARP, TCP, IPv6, and ICMP were extracted with more missing values, and further dataset preprocessing is required to fill in the missing values.

Feature preprocessing involves processing the dataset into a standard format that can be input into the model. In this paper, we used the $fillna()$ function in the pandas library to fill in missing values, labeled packets based on the source IP address, and performed a difference operation on timestamp features.

Feature selection is to exclude invalid features and select the ones that have the highest correlation with the labeled columns. On the one hand, invalid features have no effect on the model or even are detrimental to improve the model performance, on the other hand, the dataset after feature selection can reduce the computing time of the model. In this paper, the $SelectKBest$ algorithm is used to select features for the model, and six features IP_ttl, TCP_sport, TCP_dport, TCP_ack , TCP_window, IP_proto are selected.

Model training is done to deploy the best performing AI models on programmable switches for traffic detection and processing purposes. In this paper, three ML algorithms, DT, RF and S-RF, are trained to obtain the best parameters for each model.

3.3 S-RF Algorithm

RF combines the features and advantages of Bagging algorithm and Random Subspace algorithm, using DTs as classifiers, in the training process, each tree is trained using only one piece of training data, and the final result is derived from the voting decision of the results of each tree. However, RF still has some problems, RF is greatly affected by features, and when there are fewer features, it cannot produce a good classification effect.

In this paper, we design a with combining statistics and RF algorithm to preclassify the packets by counting the protocol types used in the packets in different attacks, and after that we use RF to classify the packets after classifying them by protocol types.

Algorithm 1 demonstrates the S-RF algorithm that $Packets$ denotes a collection of packets, and p_i represents a packet, and $Class$ represents the classification result of a packet. First of all,

Algorithm 1 S-RF

InPut: *Packets*
OutPut: *Class*

function ExtractFeatures(*Packets*)
    $dataset_n$ is empty data sets for different protocol packets.
    for p_i in *Packets* **do**
        if 'UDP' in p_i **then**
            p_i *storage in* $dataset_1$
        else if 'TCP' in p_i **then**
            p_i *storage in* $dataset_2$
            .....
        end if
    end for
    return $dataset_j$
end function

function RandomForest($dataset_j$)
    A DecisionTree_n() function belongs to RF algorithm.
    $i \leftarrow 0£\neg j \leftarrow 1$
    for do$i$ in $range(1, m)$
        for p_i in $dataset_j$ **do**
            $classification_1 \leftarrow$ DecisionTree_1(p_i)
            $classification_2 \leftarrow$ DecisionTree_2(p_i)
            .....
            $classification_n \leftarrow$ DecisionTree_n(p_i)
            $Class \leftarrow int((classification_1 + classification_2 + classification_n + 0.5)/n)$
            $i \leftarrow$i+1
        end for
        $j \leftarrow$j+1
    end for
    return *Class*
end function

when a packet enters into a intelligent network element, the protocol type of its transport layer is first judged, and the function is utilized to $ExteractFeatures()$ to categorize the packet into the corresponding $dataset_j$. After that, the random forest function is utilized to classify the traffic in the dataset. The random forest function contains n decision trees, and each decision tree classifies the packet p_i and then vote on the classification results of each decision tree using the averaging method to get the final classification result of the packet *Class* .

3.4 Bmv2-based INE

In this paper, BMv2 software is used for the deployment of the AI model architecture, and the role of the controller is realized by embedding the model parameters in the P4Runtime.BMv2 classifies the packets passing through the intelligent network element based on the rules issued in the P4Runtime, and realizes the detection and processing of DDoS attacks. The structure of the intelligent network element is shown in Figure 2.

The P4Runtime acts as a controller in the control plane, issuing rules through the API

interface to determine where the next table or egress is for traffic judged to be of a different type. At the same time, it exchanges control signals with the data plane composed of BMv2, realizing the control of P4Runtime over BMv2 and adjusting the rules according to the feedback from the data plane.

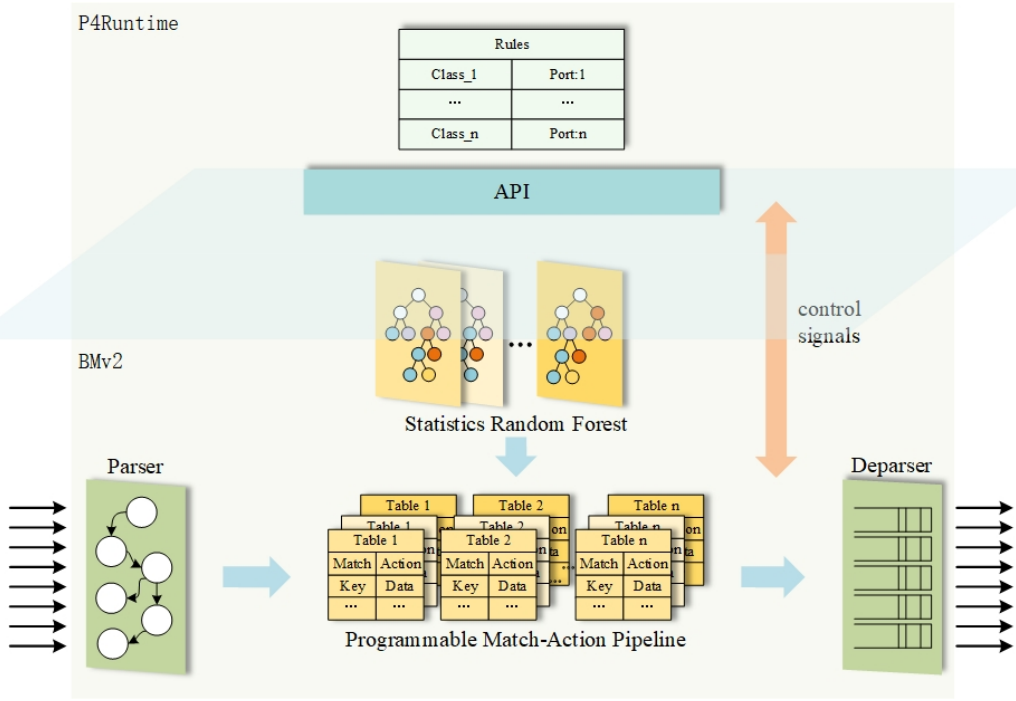

Figure 2: Intelligent network element structure

BMV2 constitutes the data plane of the intelligent network element and consists of a parser, a programmable Match-Action pipeline and an inverse parser. The parser can parse out the packet header and input the packet into the programmable Match-Action pipeline.The programmable Match-Action pipeline generated by the S-RF algorithm structure can process the packet. In the programmable Match-Action pipeline, each set of tables corresponds to a DT model; and each table corresponds to a branch node; the Match of each table is a feature, and the action corresponds to the next leaf node or branch node in the DT. The inverse parser partially repackages the obtained data to be sent to different ports to achieve traffic classification.

4 Experiments and Results

This section first describes the experimental configuration of this paper and provides a detailed description of the types of attacks on the experimental dataset used in this paper. Then evaluation Indicators used in this paper for evaluating INE are described and the results are analyzed.

4.1 Experiment Configuration

In this paper, we use Dell PowerEdge R720 with VMware ESXI virtual platform to build the overall environment of the experiment, and use VMware vSphere Client client to connect and manage it on a personal computer. A virtual machine with Ubuntu 18.04 system is used on the virtual platform. In this paper, the programmable software switch of BMv2 was deployed in Mininet software. The network topology environment as shown in Figure 3 is simulated to achieve the detection of multiple types of DDoS attacks. The attacking and attacked hosts are listed in the attack domain and attacked domain in Figure 3. In addition, normal traffic with source addresses from 23.1.0.20-23.1.0.29 is included in the attack domain.

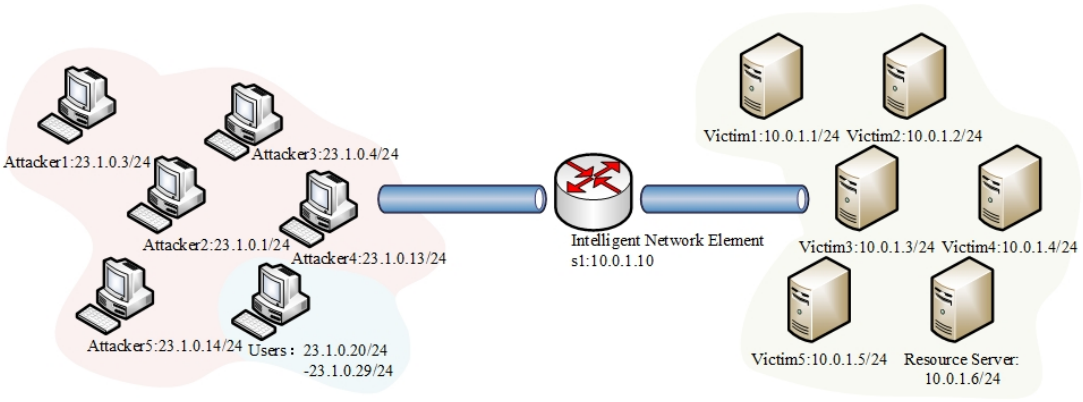

Figure 3: Network topology

Collection time	Source IP	Type of traffic	Protocol	Note
2021.5.11 20.00-23.50 &2021.5.13 15:00-2021.5.19 12:00	23.1.0.3	Bonnet	TCP	Ares, BYOB, IRC, Zeus, Mirai
2021.5.22 15:31-15:56	23.1.0.1	AL DDoS	TCP	CC
2021.5.22 20:14-20.25	23.1.0.4	DRDoS	UDP	Memcached, Chargen, NTP, SSDP, SNMP, TFTP
2021.5.23 11:08-11:11	23.1.0.13	TL DDoS	UDP&TCP	ACK, UDP
2021.5.23 15:35-16:15	23.1.0.12	LDDoS	TCP	Shrew
Full time	23.1.0.20-23.1.0.29	Begin	TCP	Includes 5G traffic

Table 1: Timetable of benchmark dataset

4.2 DDoS Dataset

The dataset for this paper uses the dataset liliMpro generated and collected in our laboratory [18] . This dataset is collected by the National Engineering Research Center for Mobile Private Networks of Beijing Jiaotong University, and contains 24 traffic types in five major categories of

DDoS, namely, AL DDoS, LDDoS, Botnet, DRDoS, and TL DDoS; the normal traffic includes 5G traffic generated by the 5G-AKA protocol. Considering the performance of INE, this paper uses some of the traffic types in the five major categories of traffic. The traffic collection time, source IP address, and traffic categories are as Table 1 shown, and the notes show the specific traffic types included in the flows used in this paper.However,Ethernet, ARP, IPv6, and ICMP protocols have been omitted from the feature selection and are therefore not displayed in the table.

4.3 Evaluation Indicators

In this paper, recall rate, precision rate, F1-Score, packet arrival rate and confusion matrix are used as evaluation indicators.

The confusion matrix measures how accurately a classifier classifies and is suitable for problems containing multiple classifiers. Table 2 shows the confusion matrix for a binary classification model, where TP (True Positive) is the positive class predicted as positive; FP (False Positive) is the negative class predicted as positive; TN (True Negative) is the negative class predicted as negative; and FN (False Negative) is the positive class predicted as negative.

Actual values \ Predicted values	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Table 2: Confusion matrix for binary classification models

Precision is the ratio of the number of samples that accurately identify the DDoS traffic as malicious encrypted traffic to the total number of samples that the model predicts to be malicious encrypted traffic; the higher the value, the more precision the model is in identifying the DDoS traffic.

$$Precision = \frac{TP}{TP + FP}$$

Recall is the ratio of the number of samples that correctly predicted the DDoS traffic as DDoS traffic to the total number of DDoS traffic samples.

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

The F1-Score is a composite of response recall and precision.

$$F1 - Score = 2 * \frac{Precision * Recall}{Precision + Recall}$$

The packet arrival rate, which reflects the ability of a intelligent network element to process packets, is the ratio of the number of arriving packets to the number of sent packets.

$$Arrival\ rate = \frac{Number\ of\ sent\ packets}{Number\ of\ arriving\ packets}$$

4.4 Evaluation Result

In this paper, the optimal parameters of the DT model and the RF model are obtained through offline training, and the models are deployed into the INE. Figure 4 shows the detection performance of DDoS attacks in INE deployed with different models. It can be seen that the DT model and the RF model have a weak detection ability in the intelligent network element, and the accuracy rate is around 65%, but the detection ability of S-RF for DDoS attacks in the intelligent network element can reach 99%. In addition, with the increase of model complexity, the packet arrival rate of the intelligent network element deployed with DT model is as high as 0.98 under the same sending rate of 1000packets/sec, while the traffic arrival rate of the intelligent network element that has gone through the deployment of RF and S-RF models is only around 82%.

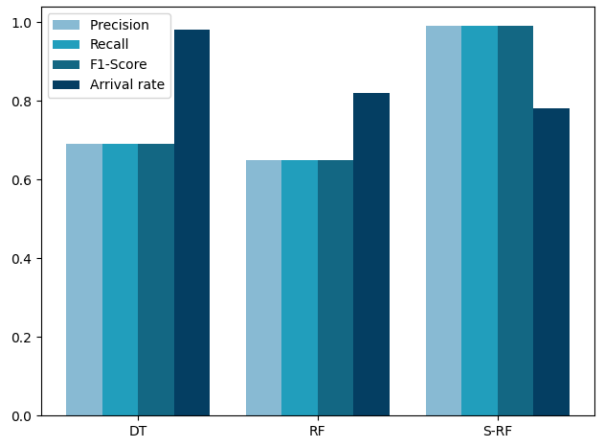

Figure 4: Comparison of the performance of three types of INE: DT, RF and S-RF

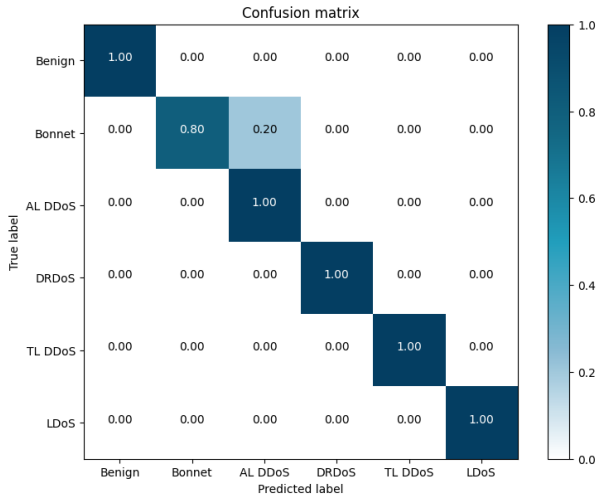

Figure 5: Confusion Matrix Detected by S-RF Algorithm in INE

Figure 5 shows the confusion matrix of the S-RF algorithm for classifying DDoS attack traffic in intelligent network elements, and it can be seen that except for a slight confusion between bot traffic and application layer DDoS attacks, the detection accuracy of all other types of traffic reaches 1. It can be seen that the preclassification of the traffic improves the model's ability to detect DDoS attacks.

5 Conclusions and Future Work

In this paper, by deploying the S-RF model in programmable switches and constructing intelligent network elements, we realize the underlying network intelligence; by detecting DDoS through intelligent network elements, we complete the classification of DDoS traffic at the network layer, improve the network security capability, and increase the DDoS attack detection accuracy rate to 99%.

Although this paper has realized the intelligence of the network to a certain extent, it still has not formed the intelligence of the network architecture, and the generalization ability of the intelligent network elements is therefore weak. In the future, we will try to utilize multiple intelligent network elements to realize the coordination of the network architecture in various aspects such as detection, arithmetic, and resources, in order to achieve a higher level of intelligence.

Acknowledgement

This paper is supported by National Key R&D Program of China under Grant No. 2018YFA0701604

References

[1] Cloudflare. Threat Report Cloudflare DDoS Trends Report. Technical report, Cloudflare, 2023.

[2] Xiang Chen, Chunming Wu, Xuan Liu, Qun Huang, Dong Zhang, Haifeng Zhou, Qiang Yang, and Muhammad Khurram Khan. Empowering Network Security With Programmable Switches: A Comprehensive Survey. *IEEE Communications Surveys & Tutorials*, 25(3):1653–1704, 2023.

[3] 6GANA. Ten Questions of 6G Native AI Network Architecture. Technical report, 6GANA, 2022.

[4] ITU-R. Future technology trends of terrestrial International Mobile Telecommunications systems towards 2030 and beyond. Technical report, ITU, 2022.

[5] Huawei. 6G: A New Journey For Wireless Communication. Technical report, Huawei, 2022.

[6] Saman Taghavi Zargar, James Joshi, and David Tipper. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. *IEEE Communications Surveys & Tutorials*, 15(4):2046–2069, 2013.

[7] Kurt Friday, Elie Kfoury, Elias Bou-Harb, and Jorge Crichigno. Towards a Unified In-Network DDoS Detection and Mitigation Strategy. In *2020 6th IEEE Conference on Network Softwarization (NetSoft)*, pages 218–226, Ghent, Belgium, June 2020. IEEE.

[8] Mahsa Nooribakhsh and Mahdi Mollamotalebi. A review on statistical approaches for anomaly detection in DDoS attacks. *Information Security Journal: A Global Perspective*, 29(3):118–133, May 2020.

[9] Zhongmin Wang and Xinsheng Wang. DDoS attack detection algorithm based on the correlation of IP address analysis. In *2011 International Conference on Electrical and Control Engineering*, pages 2951–2954, Yichang, China, September 2011. IEEE.

[10] Loic D. Tsobdjou, Samuel Pierre, and Alejandro Quintero. An Online Entropy-Based DDoS Flooding Attack Detection System With Dynamic Threshold. *IEEE Transactions on Network and Service Management*, 19(2):1679–1689, June 2022.

[11] Huan Lin, Shoufeng Cao, Jiayan Wu, Zhenzhong Cao, and Fengyu Wang. Identifying Application-Layer DDoS Attacks Based on Request Rhythm Matrices. *IEEE Access*, 7:164480–164491, 2019.

[12] Kshira Sagar Sahoo, Bata Krishna Tripathy, Kshirasagar Naik, Somula Ramasubbareddy, Balamurugan Balusamy, Manju Khari, and Daniel Burgos. An Evolutionary SVM Model for DDOS Attack Detection in Software Defined Networks. *IEEE Access*, 8:132502–132513, 2020.

[13] Amjad Alsirhani, Srinivas Sampalli, and Peter Bodorik. DDoS Detection System: Using a Set of Classification Algorithms Controlled by Fuzzy Logic System in Apache Spark. *IEEE Transactions on Network and Service Management*, 16(3):936–949, September 2019.

[14] Ahmad Zainudin, Love Allen Chijioke Ahakonye, Rubina Akter, Dong-Seong Kim, and Jae-Min Lee. An Efficient Hybrid-DNN for DDoS Detection and Classification in Software-Defined IIoT Networks. *IEEE Internet of Things Journal*, 10(10):8491–8504, May 2023.

[15] Damu Ding, Marco Savi, Federico Pederzolli, Mauro Campanella, and Domenico Siracusa. In-Network Volumetric DDoS Victim Identification Using Programmable Commodity Switches. *IEEE Transactions on Network and Service Management*, 18(2):1191–1202, June 2021.

[16] Wenji He, Yifeng Liu, Haipeng Yao, Tianle Mai, Ni Zhang, and F. Richard Yu. Distributed Variational Bayes-Based In-Network Security for the Internet of Things. *IEEE Internet of Things Journal*, 8(8):6293–6304, April 2021.

[17] Yongyi Cao, Hao Jiang, Yuchuan Deng, Jing Wu, Pan Zhou, and Wei Luo. Detecting and Mitigating DDoS Attacks in SDN Using Spatial-Temporal Graph Convolutional Network. *IEEE Transactions on Dependable and Secure Computing*, 19(6):3855–3872, November 2022.

[18] Li Lijuan, Li Yingzhi, Li Tianqi, Shen Qi, and Li Man. GitHub-LiliMpro/source_dataset, August 2022.