BCDM: An Early-Stage DDoS Incident Monitoring Mechanism based on Binary-CNN in IPv6 Network

Yufu Wang, Xingwei Wang*, Qiang Ni, *Senior Member, IEEE,* Wenjuan Yu, Min Huang

***Abstract*—The rapid adoption of IPv6 has increased network access scale while also escalating the threat of Distributed Denial of Service (DDoS) attacks. By the time a DDoS attack is recognized, the overwhelming volume of attack traffic has already made mitigation extremely difficult. Therefore, continuous network monitoring is essential for early warning and defense preparation against DDoS attacks, requiring both sensitive perception of network changes when DDoS occurs and reducing monitoring overhead to adapt to network resource constraints. In this paper, we propose a novel DDoS incident monitoring mechanism that uses macro-level network traffic behavior as a monitoring anchor to detect subtle malicious behavior indicative of the existence of DDoS traffic in the network. This behavior feature can be abstracted from our designed traffic matrix sample by aggregating continuous IPv6 traffic. Compared to IPv4, the fixed-length header of IPv6 allows more efficient packet parsing in preprocessing. As the decision core of monitoring, we construct a lightweight Binary Convolution DDoS Monitoring (BCDM) model, compressed by binarized convolutional filters and hierarchical pooling strategies, which can detect the malicious behavior abstracted from input traffic matrix if DDoS traffic is involved, thereby signaling an ongoing DDoS attack. Experiment on IPv6 replayed CIC-DDoS2019 shows that BCDM, being lightweight in terms of parameter quantity and computational complexity, achieves monitoring accuracies of 90.9%, 96.4%, and 100% when DDoS incident intensities are as low as 6%, 10%, and 15%, respectively, significantly outperforming comparison methods.**

***Index Terms*—DDoS monitoring, Binary-CNN, Traffic matrix, Network behavior, IPv6 network.**

I. Introduction

TODAY, the state of global cybersecurity is deteriorating seriously. According to the Cybersecurity Forecast 2024 Report[1], the number of global cybersecurity incidents has increased 11 times in the past eight years, where the proportion of DDoS (Distributed Denial of Service) attacks is as high as 50.2%. DDoS attack is a type of attack against network services, where attackers send massive traffic to the target by commanding the distributed botnets, in order to exhaust the service capacity of the victim host or server, making it unable to receive or respond to normal service requests from legitimate users. With the increasing spread on a global scale, DDoS has become one of the common network attacks that is large-scale, harmful and difficult to prevent.

As a next-generation Internet technology, IPv6 is rapidly being popularized worldwide, such as France 73.47%, India 68.88%, Germany 65.93%, United States 49.17%, and United Kingdom 43.71%[2]. Concurrently, more and more Internet service providers are continuously increasing the business on IPv6 networks. As shown in Fig. 1, in 2023, the proportion of users providing services through pure IPv6 network of Google has reached 42.42%[3] and Facebook 37.27%[4] , with these numbers expected to rise at an accelerating pace in the future.

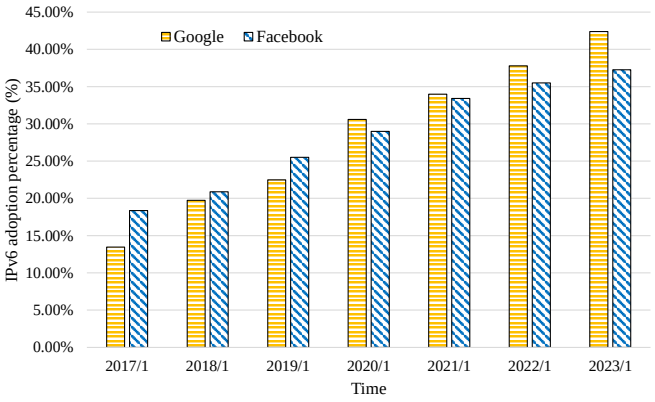


Fig. 1: IPv6 business of Google and Facebook[3,4]

However, as shown in Fig. 2, even though IPsec is introduced to enhance IPv6 security, there are still severe security risks, with DDoS attacks representing the biggest threat as 68%[5]. Alarmingly, in March 2018, internet engineers encountered the first DDoS assault relying solely on IPv6. The Report[6] indicate that over the past few years, a major rise in the share of malicious DDoS traffic carried IPv6 protocol to the tune of 600%. In the future, the larger address range and greater access volume brought about by the IPv6 continued adoption will inevitably be accompanied by more intense DDoS attacks[7]. Up to now, the peak traffic of DDoS attacks

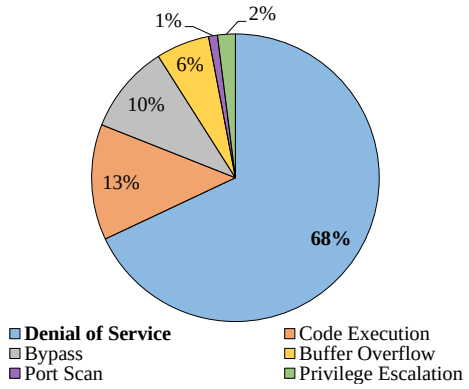


Fig. 2: IPv6 vulnerability classes[5]

Y. Wang, X. Wang, and M. Huang are with the Northeastern University, China, e-mail: (wangyufu_neu@outlook.com).

Q. Ni and W. Yu are with Lancaster University.

on record has reached 1.45 Tbps. Often, by the time the victim is aware that they are being attacked, the already evolved massive DDoS traffic is challenging to mitigate[8].

Consequently, considering the forward-looking nature of IPv6 network DDoS defense, it is not solely about filtering DDoS attack traffic but, more importantly, providing early warnings when DDoS incidents occur. The earlier the intervention, the more time can be gained for deploying defense measures, thereby reducing the pressure. This is the DDoS incident monitoring problem targeted by this paper, which will be more valuable as the threat of DDoS increases[9]. However, this endeavor faces multifaceted challenges: a. Real-time processing: The monitoring system requires efficient traffic processing and analysis capabilities to respond in real-time to sudden DDoS attacks; b. High Sensitivity: Higher sensitivity can detect signals at the onset of abnormal behavior and trigger alarms promptly, depending on the selected monitoring features and methods; c. Resource constrains: As a long-running task, monitoring must maintain lightweight overhead within the limited computing and storage resources of network devices[10]; d. Attack diversity: The diversity and continuous evolution of DDoS attacks necessitate that the monitoring system be comprehensive and flexible in analyzing traffic patterns; and e. Analytical ability: Amidst the exponential growth of network scale and traffic volume, the monitoring system requires robust data analysis capabilities, including the application of big data analysis and deep learning methods.

While IPv6 network introduce additional challenges to the research and deployment of DDoS monitoring[11]. The vast address space of IPv6 places higher demands higher efficiency of traffic parsing in monitoring system, and new protocols like ICMPv6 may become new attack vectors that need to be considered. Despite being launched many years ago, IPv6 still lacks mature security tools and protection solutions compared to IPv4[12], especially evident in the scarcity of IPv6 DDoS datasets. In contrast to the widely used IPv4 DDoS datasets, such as CIC-DDoS2019 and CAIDA, the IPv6 environment lacks corresponding widely recognized data resources, significantly impeding the progress of defense research in this field.

To address these challenges, we propose a novel DDoS incident monitoring mechanism in IPv6 networks. By designing a two-dimensional traffic matrix sample to aggregate packet-level data of continuous network traffic, a sensitive macroscopic network behavior feature is abstracted as the anchor point for our monitoring, which can reflect the existence of DDoS traffic in the network. Compared to traditional flow-level monitoring, packet-level data, with its complete payload and header information, can serve as a more detailed data source for real-time, flexible, and resource efficient network activity monitoring[13]. At the heart of our mechanism lies a deep learning DDoS monitoring model (called BCDM) based on the binary convolutional neural network[14], which not only possesses robust feature extraction and analysis capabilities, but has also been meticulously optimized for computational efficiency and parameter reduction, ensuring a lightweight monitoring overhead. Upon training, even with sporadic DDoS traffic interspersed in the network, BCDM can perceive the subtle malicious behavior in the network traffic through the traffic matrix input, thus providing sensitive early warning of the start of a DDoS incident. The main contributions are summarized as follows:

- To enhance the comprehensiveness and flexibility of monitoring application, our mechanism considers three primary traffic protocols in IPv6 network: ICMPv6, UDP, and TCP, thereby covering most DDoS attack types. Unlike IPv4, the unique fixed-length format of IPv6 header enables us to design a more efficient traffic preprocessing strategy, reducing memory overhead and improving processing efficiency. The hexadecimal packet fields, as our data source for monitoring, are directly parsed from raw traffic through traversal, which improves the real-time performance and perform scalability under the growth trend of IPv6 network scale and complexity.
- Innovatively, we propose a macro network traffic behavior feature as our monitoring anchor point, which is derived through the aggregation of continuous network traffic data within our meticulously designed 100x82 traffic matrix sample. On this basis, we construct a BCDM deep learning model as the decision-making core of monitoring, capable of detecting the subtle malicious behavior caused by the existence of low-rate DDoS traffic in the network, thereby inferring the early start of DDoS incidents. To meet the resource constrains of network devices in monitoring deployment, BCDM is designed to be lightweight, utilizing binarized convolutional filters and hierarchical pooling strategies.
- In our experiment, to address the absence of IPv6 DDoS public datasets, we first attempt to construct our reliable IPv6-DDoS traffic sources by utilizing NAT 4to6 Jool tool[15] to perform IPv6 replay of IPv4 CIC-DDoS2019[16] public set within self-built IPv6-LAN topology, situated in CERNET2[17] IPv6 environment. Then, referring to the DDoS intensity indicator—defined as the percentage of DDoS traffic rate to the total network rate-we construct traffic matrix samples under varying intensities, controlling the variables through the assumption of uniform distribution to ensure the stability of training and testing. On this basis, BCDM is evaluated in terms of accuracy, precision, recall, F1-score, ROC(Receiver Operating Characteristic), and AUC(Area Under ROC Curve) under different DDoS intensity, that outperforms with compared methods.

The rest of this work is organized as follows. Section II reviews the related work. Section III introduces the system framework. Sections IV and V illustrate the IPv6 traffic preprocessing strategy and BCDM monitoring model design in detail. Our experiment topology, setup, and results are presented in Section VI. The conclusion is in Section VII.

II. Related work

In related work, we first analyze the DDoS traffic detection researches based on deep learning, that is, how to distinguish whether a single packet or flow is DDoS traffic, which inspires us to apply the deep learning concept into DDoS monitoring scenario. Secondly, we introduce the existing DDoS monitoring researches, outline the drawbacks and describe the

improvement aspects of our proposed mechanism. Finally, we compare the different compression methods of CNN models to demonstrate the adaptability of binary-CNN in DDoS monitoring scenario.

A. DDoS traffic detection based on deep learning

DDoS traffic detection aims to distinguish DDoS attack traffic from normal traffic by analyzing network traffic characteristics, and in recent years, the introduction of deep learning technology has gradually increased. Abdallah et al. [18] employed a Deep Neural Network (DNN), achieving a significantly lower false alarm rate compared to traditional entropy-based methods. Cil et al. [19] demonstrated that DNNs can swiftly and accurately detect DDoS in small sample sets. Aydin et al. [20] introduced LSTM-CLOUD, utilizing Long Short-Term Memory (LSTM) networks for DDoS detection and prevention in cloud networks. In IPv6 network, Manickam et al. [21] proposed an ICMPv6 DDoS detection framework (v6IIDS) based on a back-propagation neural network. Meanwhile, the CNN model of the basis of our work is also used in DDoS traffic detection. Shieh et al. [22] built a Convolutional Neural Network (CNN) construction featuring geometrical metric (CNN-Geo) to utilize deep learning techniques to enhance DDoS attack detection accuracy. Shalaka et al. [23] proposed an intelligent intrusion detection system (IDS) using a CNN, i.e., HetIoT-CNN IDS to solve the DDoS traffic detection of the heterogeneous IoT (HetIoT). Yousif et al. [24] combined mininet, Ryu controller, and one dimensional-CNN to detect and mitigate DDoS attacks in SDN environments. Sharma et al. [25] proposed an efficient deep learning-based CNN-Bidirectional LSTM for the DDoS detection, where the LSTM is used to extract features for the classification of CNN part.

The aforementioned studies fully demonstrate the ability of neural networks to identify DDoS traffic features, including the wide identification types, high accuracy and strong time series perception[26], especially the CNN model. In our DDoS monitoring study, we attempt to extend this DDoS feature recognition capability by designing a CNN-based BCDM mechanism, enabling it to monitor the occurrence of DDoS incident through identifying the malicious changes of network traffic behavior abstracted by traffic matrix view.

B. DDoS incident monitoring works

Different from traffic detection, the DDoS incident monitoring aims to dynamically monitor the macro characteristics of the network to more sensitively perceive the occurrence of DDoS attacks in a normal network. The early classic work, Yuan et al. [27] introduced a macro-monitoring approach using cross-correlation at network observation points to detect traffic pattern changes indicative of DDoS attacks. As the intensity of DDoS attacks increases, monitoring tasks are increasingly being studied as triggers for initiating DDoS defense, thereby enriching defense preparations and reducing the pressure of mitigation. Segura et al. [28] offered a lightweight, efficient detection method based on change point analysis to identify anomalies in packet delivery rates and overhead. Entropy value is a more common monitoring index, Li et al. [29] presented an early detection method in SDN networks using φ-entropy to enhance traffic feature distinctions. Ahalawat et al. [30] proposed a DDoS detection technique based on Renyi Entropy with Packet Drop (REPD) where packets drop method is used for the purpose of mitigation. Aladaileh et al. [31] devised an entropy-based method for DDoS detection in SDN, aiming to increase accuracy for high-rate attacks and lower false positives in varied scenarios. There are also methods based on machine learning or deep learning. Xie et al. [32] introduced an anomaly detector using a hidden semi-Markov model to capture dynamic access distribution changes for attack monitoring. Zhou et al. [33] proposed an online Internet traffic monitoring framework based on Spark Streaming and Flink for real-time DDoS monitoring. Feng et al. [34] combined Generalized Network Temperature with deep learning to enhance predictions and classifications of network congestion, improving DDoS early warnings. Kirtas et al. [35] utilized a photonic neuromorphic lookaside accelerator for real-time inspection of DDoS attack indicators, such as port-scanning operations.

To sum up, as DDoS monitoring has advanced, the continuous development of new methods such as macro-observation, entropy-based statistical, clustering, Markov chain, and neural network enable detectors to identify and respond to the occurrence of DDoS incidents at an early stage with lower intensity; however, the application of deep learning is still relatively simple. In this paper, we further expand the strong traffic feature recognition ability of deep learning model, and design a lightweight BCDM model to improve the performance of DDoS incident monitoring.

C. Compression strategies on CNN model

CNN models are gradually being applied in network traffic analysis, where different compression strategies are used to reduce their overhead. For instance, Saiyed et al. [36] employed parameter pruning to lower memory and processing requirements for DDoS attack detection, while Li et al. [37] utilized tensor decomposition for reconstructing binary adjacency matrices. Wang et al. [38] also compressed MSSTRNet into the lightweight LENet using knowledge distillation, and LEE et al. [39] applied low-rank tensor decomposition and lossy tensor compression to reduce training memory usage. Additionally, He et al. [40] showed that adding depthwise separable convolution (DSC) to PyConv reduces network complexity, Lu et al. [41] used dilated convolution in anonymous traffic recognition to expand the receptive field without increasing parameters or computational complexity, and Le et al. [42] demonstrated that binary-CNN can significantly reduce model size and computational cost.We summarize the different aspects of these compression methods in Table I. In fact, effective compression methods should align with the requirements of their specific working scenarios. Therefore, after analyzing above compression strategies, we choose binary-CNN as the main compression strategy for our CNN monitoring core, as its advantages can match the requirements of DDoS monitoring scenario. a. Large scale data monitoring: Binary-CNN significantly reduces the storage

TABLE I: Detail of CNN model compression strategies

Strategy	Working principle	Benefits	Drawbacks or limitation
Pruning[36]	Reduce redundant parameters and connections.	Reduce parameter count, maintain performance, improve reasoning speed.	Additional iteration steps, increase model complexity, unstable performance.
Tensor Decomposition[37]	Decompose high-dimensional tensors into several low dimensional tensors.	Reduce parameter count, reduce computational complexity, suitable for large-scale models	Complex decomposition algorithm, data dependency, complex parameter tuning.
Knowledge Distillation[38]	Transfer knowledge from large teacher model to small student model.	Improvement of small model, high flexibility, reduce inference time.	Increase training complexity, teacher model dependency, complex parameter tuning.
Low-rank Factorization[39]	Decompose convolutional kernels into smaller matrices.	Reduce parpmeter count, reduce computation complexity, compatible with multiple models.	Complex decomposition algorithm, data dependency, complex parameter tuning.
Depthwise Separable[40]	Decompose standard convolution into depthwise and pointwise convolutions.	Reduce computational load, reduce parameter count, easy to combine.	Need more layers, performance drop, debugging complexity.
Dilated Convolution[41]	Insert holes (zeros) between convolutional kernel elements.	Expand the receptive field, increase of efficiency, maintain resolution.	Gradient instability, complex implementation, task limitations.
Binary-CNN[42]	Restrict weights and activations to binary values (-1 and 1).	Extreme compression rate, high efficient bitwise computation, simple structure with low overhead.	Performance drop, increase training complexity, specific hardware support.

and computational requirements through extreme compression rates, enabling efficient large-scale data monitoring. b. Real-time Processing: The efficient bitwise computation ensures that binary-CNN allows for rapid processing of high-throughput data streams, ensuring timely response of DDoS attacks, and enabling swift countermeasures. c. Diversified deployment: The simple model structure with low overhead enables binary-CNN to be flexibly deployed on various resource constrained edge devices, achieving broad monitoring coverage to meet the diverse needs of DDoS monitoring. On this basis, we further deepen and optimize the binary-CNN model structure, achieving our DDoS incident monitoring mechanism.

III. System Framework

With the development of DDoS attacks, although the emerging attack methods are complex and changeable, such as reflection and amplification methods, "traffic" is still a necessary component of DDoS attacks. As a result, starting from the analysis of network traffic, we implement the monitoring of early DDoS incidents in IPv6 environment based on packet-level traffic information. We design a BCDM monitoring core, which is a lightweight deep learning model based on Binary-CNN, and design a traffic matrix sample as model input to abstract traffic behavior pattern by aggregating continuous traffic information. When DDoS traffic exists, BCDM can perceive the maliciousness of the traffic behavior pattern. The workflow of our DDoS monitoring mechanism is shown in Fig. 3. First, the monitor device dynamically captures ongoing network traffic from IPv6 network and stores it directly into the pcap traffic file format. Secondly, the traffic will next be preprocessed; based on the fixed-length property of IPv6 packet headers, we quickly parse the pcap file, extract the hexadecimal header fields, and aggregate the information from every 100 packets into a traffic matrix. Finally, traffic matrices will be input into our designed BCDM model as samples, which can directly detect the malicious abstracted traffic behavior, that is, the existence of DDoS traffic in the network, thereby inferring the start of a DDoS incident.

Next, we will introduce the above workflow in detail.

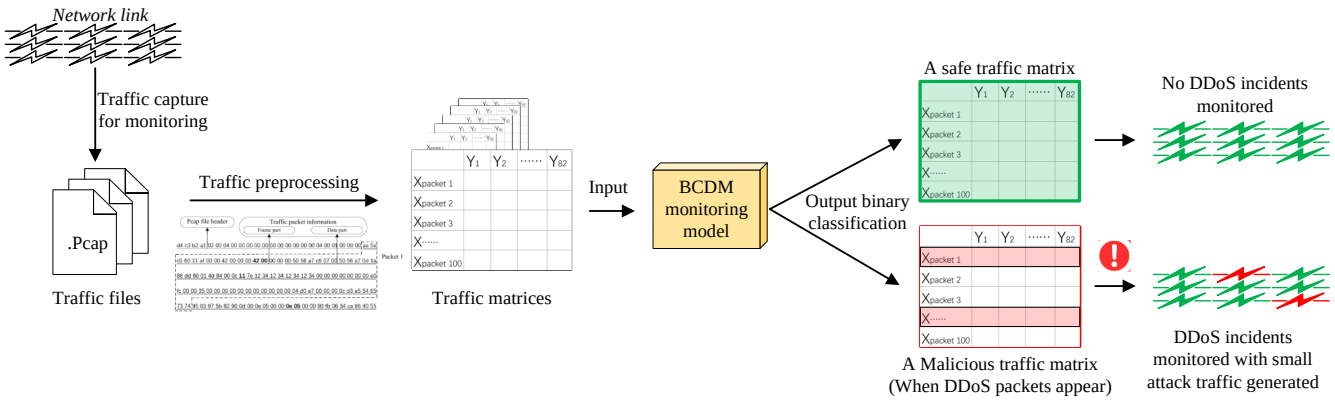


Fig. 3: Workflow of our DDoS monitoring mechanism

IV. IPv6 TRAFFIC PREPROCESSING STRATEGY

A. Extraction of IPv6 packet header information

The purpose of the IPv6 traffic preprocessing strategy is to convert network-captured IPv6 data into samples suitable for our deep learning monitoring model. In our work, we use wireshark to capture IPv6 traffic and store into pcap files as sources. Then, based on the fixed-length characteristic of IPv6 traffic header, we directly parse the hexadecimal fields stored in pcap as sample data, omitting the traditional traffic translation steps to improve the traffic preprocessing efficiency.

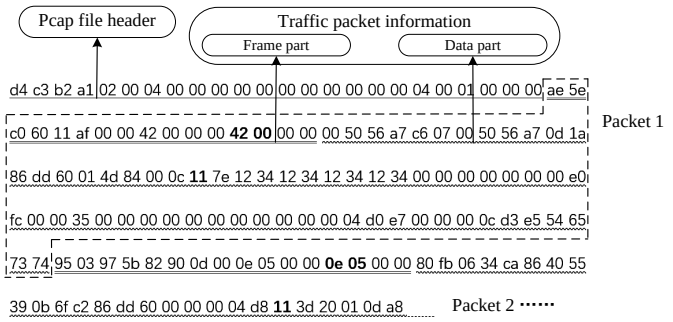


Fig. 4: pcap storage format

As shown in Fig. 4, a pcap file begins with a file header, which has a set length of 24 bytes and records the time interval and packet numbers captured in the current pcap file. Then there are packets stored in sequence, where the part surrounded by a dotted line is the first packet information, consist of two parts: the frame part, denoted by double underlines, which is used to record the arrival time, elapsed time, capture length, and etc., has a fixed length of 16 bytes; and the data part, denoted by wavy underlines, which is used to record the header fields and payload data, has a variable length.

On the basis of this, the process of our parsing pcap is as shown in Fig. 6. First, the 24-byte pcap file header is skipped. Then, the 13th to 14th bytes of the frame part store the length of current packet in reverse order, as ”0042” in Fig. 4, that is, 66 bytes. Therefore, we can get the data part content by extracting 66 bytes from the end of frame part as well as the starting position of the next packet's frame part. Finally, we will determine the packet's protocol type according to the 21st byte in the data part. For instance, the bold ”11” in packet 1 indicates that this protocol type is ICMPv6. In this way, combined with the fixed-length properties of different protocols headers in IPv6 traffic, the complete header fields can be parsed out. In Fig. 5, we analyze the header fields and corresponding byte positions of ICMPv6, TCP and UDP packets commonly encountered in IPv6 network DDoS monitoring, including the common fields (ETH and IPv6) and the special fields of different protocol types[43].

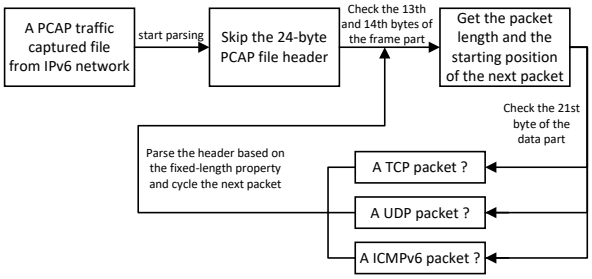


Fig. 6: The process of pcap prasing

As indicated in Table II, the three DDoS attack methodologies we address occur at the foundational levels of network communication, namely the network and transport layers. Hence, the characteristics of these attacks are primarily evident in the header fields of the network and transport layer protocols. Our information parsed in Fig. 5 can furnish ample data to identify potential attack behaviors, highly relevant for DDoS monitoring. Additionally, these three protocols cover most known DDoS attack types, the all considered can enhance the comprehensiveness and effectiveness of our monitoring.

TABLE II: ICMPv6, TCP, and UDP DDoS attack

DDoS protocal type	Layer	Typical DDoS attack
ICMPv6	Network layer	IPv6 Ping flood Attack, Smurf Attack, Neighbor Discovery Attack
TCP	Transport layer	SYN Flood, ACK Flood, RST Flood, TCP Connection Flood
UDP	Transport layer	UDP Flood, UDP Amplification, Fraggle Attack

IPv6 packet header = Common header + Special header = (Eth header Byte 1-14 + IPv6 header Byte 15-54) + ICMPv6 header (Byte 55-END) or UDP header (Byte 55-END) or TCP header (Byte 55-END)

Common	Eth header			IPv6 header								
Byte	1-6	7-12	13,14	15(1/2)	15(2/2)	16(1/2)	16(2/2) -18	19,20	21	22	23-38	39-54
Name	Eth. dst	Eth. addr	Eth. type	IPv6. ver	IPv6. tclass.dscp	IPv6. tclass.ecn	IPv6. flow	IPv6. plen	IPv6. nxt	IPv6. hlim	IPv6. src	IPv6. dst

Special	ICMPv6					
Byte	55	56	57-58	59-60	60-62	63-END
Name	type	code	checksum	identifier	sequence number	payload

Special	UDP				
Byte	55-56	57-58	59-60	61-62	63-END
Name	srcport	dstport	length	checksum	payload

Special	TCP									
Byte	55-56	57-58	59-62	63-66	67(1/2)	67(2/2)-68	69-70	71-72	73-74	75-END
Name	srcport	dstport	seq	ack	hdr_len	flags	window size value	checksum	urgent pointer	payload

Fig. 5: Header fields and byte positions of ICMPv6, TCP, and UDP in IPv6 network packet

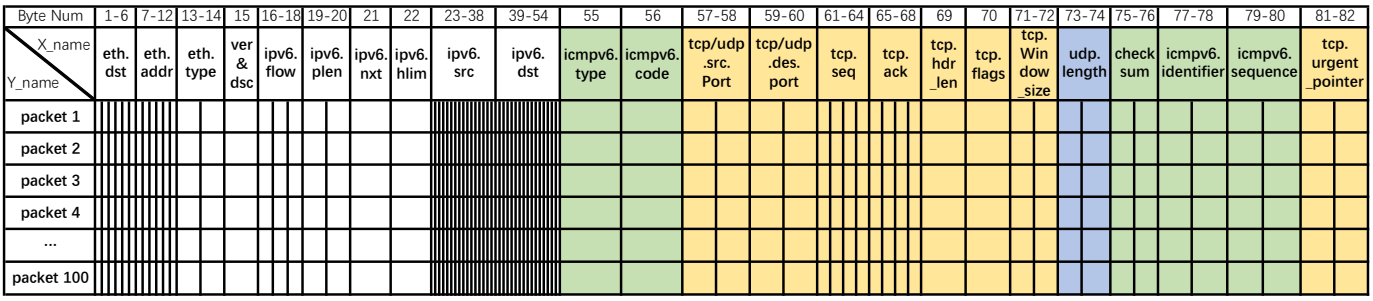


Fig. 7: The format of traffic matrix sample

B. Design of the traffic matrix input sample

In this paper, we monitor DDoS incidents by determining whether sporadic DDoS traffic appears in the network; however, under the background of normal network traffic rate, the traditional methods of distinguishing packets one by one is not feasible, which is not only inefficient but also result in significant overhead. Therefore, as shown in Fig. 7, we design a more effective two-dimensional traffic sampling unit to aggregate traffic sets, leveraging the overall malicious traffic posture characteristics induced by the presence of DDoS traffic to realize a high-efficiency solution. Based on the header analysis in Fig. 5, we finally set up 82 bytes in X-axis, where X_{1-54} are the common header fields. The remaining parts are the unique fields, where ICMPv6 fills in $X_{55,56,75-80}$, TCP fills in $X_{57-72,81-82}$, UDP fills in $X_{57-60,73-76}$. When filling, we first fill the header fields into the corresponding positions of each row according to the protocol type of each packet, and finally fill the empty fields with 0. And the matrix's row count is set to 100, representing a collection of 100 packets per matrix, is rationalized by several key considerations: a. After investigating the input aspect ratio of related DDoS convolution detection, such as Shaaban[44] 1.17, Kim[45] 0.7, and Hussain[46] 1, the matrix with an aspect ratio around 1 can better match the square convolution kernel receptive field. b. Satisfy the adjustment of the number of DDoS packets within the matrix in accordance with the percentage-based DDoS intensity value we set in this paper. c. The size of 100x82 strikes a good balance between model performance and overhead, even though larger input sizes that provide more contextual detail could potentially increase accuracy. However, doubling the row count to 200 would lead to 424.90% and 133.12% increase of computational load and memory overhead for our BCDM model. Given that the network environment is inherently constrained in terms of computational resources, the current input size of 100x82 is likely more optimal.

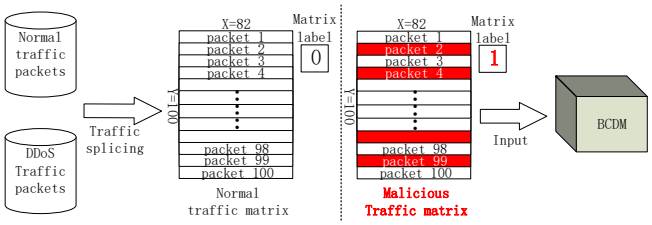


Fig. 8: Two categories of the traffic matrix samples labels

Further, we build our traffic matrix data set, that consist of <traffic matrix, matrix label>. As shown in Fig. 8, the traffic matrix samples are divided into two categories, namely, the normal traffic matrix (label 0) containing all normal packets, and the malicious traffic matrix (label 1) mixed with DDoS packets. After training, our BCDM model can monitor network traffic with a large matrix view, sense the malicious behavior changes caused by DDoS traffic involved, and directly make coarse-grained and rapid judgments on the existence of DDoS traffic in the network. Meanwhile, by adjusting the mixed percentage of DDoS packets in the traffic matrix sample, we can simulate the network traffic behavior with different DDoS intensities, so as to train and test the monitoring performance of BCDM model in the experiment.

V. BCDM: The Binary Convolution based DDoS Monitoring model

In our DDoS monitoring mechanism, we take the traffic matrix as input sample to dynamically monitor the occurrence of DDoS incidents in the network. To realize this matrix analysis vision, we first introduce the convolutional neural network to design the base structure (Base-CNN) of monitoring core. On this basis, considering the normalized operation characteristic of monitoring, we further introduce the binary convolution kernel method, combined with model compression methods such as global average pooling, to build a more efficient and low-overhead deep learning decision-making core, called BCDM, whose structure is shown in Table III.

TABLE III: The structure of Base-CNN and BCDM

Base-CNN			**BCDM**		
Input	100*82		Input	100*82	
Layername	Output size	7 layers	Layername	Output size	12 layers
Conv_1_x	94*76,2	7*7,2,stride 1	Conv_1	94*76,4	7*7,4,stride 1
Conv_2_x	94*76,4	5*5,4,stride 1 5*5,4,stride 1	Binary_conv1_x	47*38,8	5*5,8,stride 1 5*5,8,stride 1 2*2 max pooling, stride 2
Conv_3_x	94*76,8	3*3,8,stride 1 3*3,8,stride 1	Binary_conv2_x	23*19,16	3*3,16,stride 1 3*3,16,stride 1 2*2 max pooling, stride 2
Flatten	57152		Binary_conv3_x	11*9,16	3*3,16,stride 1 3*3,16,stride 1 3*3,16,stride 1
Dense	2		GAP	16	global average pool, stride 2
			Dense	2	

The Base-CNN model contains 7 layers, with a total of 113,566 parameters, including one 7*7, two 5*5 and two 3*3 convolutional layers, one flatten layer and one dense layer for output two types classification. During the design process of Base-CNN, we start with two convolutional layers structure

and gradually increase the convolutional layers, depth, and channels to improve the identification ability; until Base-CNN is able to achieve 98.3% accuracy on traffic matrices with 10% DDoS traffic intensity, we determine the above model structure and optimize it as a the starting point.

In Fig. 9, we compare the structure diagram of our BCDM model and Base-CNN. As present in Table III, BCDM model consists of a total of 12 layers, starting from the input layer, including one normal 7*7 convolutional layer(conv_X), two 5*5 and four 3*3 binarized convolutional layers(binary_convX), one Global average pooling (GAP) layer, and one softmax layer for output binary classification, where every two binarized convolutional layers are followed by a max pooling layer. When compared to Base-CNN, the optimization of the BCDM model is primarily reflected in three aspects:

a. Compression of fully connected parameters: Firstly, Base-CNN contains 98.5% of fully connected parameters. Therefore, to compress the model, we use GAP to replace the original flatten layer, reducing the number of variables input to the softmax layer from 57152 to 16. In this way, the model parameter quantity decreases from 113566 to 11114.

b. Compression of convolutional parameters: Secondly, we further study the lightweight of the convolutional layer part of the model. We introduce the idea of binarized parameter to form the binarized convolutional layer[42] (Binary_conv), that can binarize the float-type convolution kernel parameters into one bit 1 or -1 through the Equation 1 deterministic binarization function.

$$X^{\mathrm{b}} = Sign(x) = \begin{cases} +1 & \text{if} \quad \mathrm{x} \geq 0 \\ -1 & \text{otherwise} \end{cases} \tag{1}$$

Algorithm 1 demonstrates our procedure for training a CNN with binary weights. First, we binarize the weight filters at each layer by computing A_{lk} and B_{lk}, where c, w, and h represent the size of channels, width, and height of kernels.

Algorithm 1. BCDM parameters training forward propagation

Input:The minibatch of inputs and targets (I,Y), cost function C($Y, \hat{Y}$), current weight W^{t} and current learning rate η^{t}.
Output: The updated weight $W^{\mathrm{t+1}}$ and updated learning rate $\eta^{\mathrm{t+1}}$
01: Binarizing weight filters:
02: **for** k=1 to L **do** (cycle from the 1st layer to the L$^{\mathrm{th}}$ layer)
03: $\quad \mathrm{A}_{lk} = \frac{1}{cwh} \left\| W_{lk}^{t} \right\|_{\ell 1}$
04: $\quad B_{lk} = Sign(W_{lk}^{t})$
05: $\quad \tilde{W}_{lk} = A_{lk} B_{lk}$
06: **END for**
07: $\hat{Y} = BinaryForward(I, B, A)$
08: $\frac{\partial C}{\partial \tilde{W}} = BinaryBackward(\frac{\partial C}{\partial \hat{Y}}, \tilde{W})$
09: $\tilde{W} = UpdateParameters(\tilde{W}_t, \frac{\partial C}{\partial \tilde{W}}, \eta^t)$
10: $\eta^{t+1} = ReduceLROnPlateau(\eta^t, t)$

Then we call the $BinaryForward$ function $(I \oplus B)A$ with the binary weights(B) and its scaling factors(A). Then, we call BinaryBackward function to compute the gradients with respect to the estimated weights and update the value with UpdateParamters function. Lastly, the learning rate will be dynamically updated with the ReduceLROnPlateau rule. In this way, the parameters can be binarized during model training. Until the last layer, the activation value will be directly output to the full connection without binarization. In addition, in this convolution model, the first convolutional layer cannot be binarized in order to preserve the matrix features. This binary convolution method can bring the following two lightweight advantages in DDoS normalized monitoring:

- Significant reduction in model size and running memory: The convolution kernel parameters of Base-CNN are stored in 32-bit floating point, while the binarized parameters can only be stored in one bit. Therefore, after GAP involved, the binarization of the convolution parameters can reduce the overall storage size and running memory overhead of the model to 1/32 of the original.

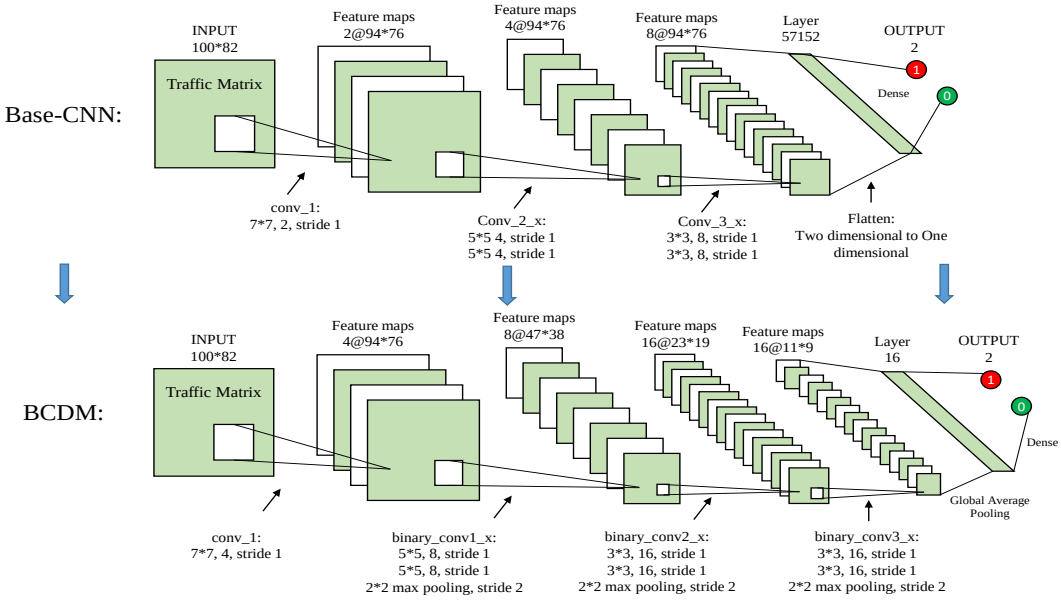


Fig. 9: Base-CNN and BCDM model structure diagram

- Significant reduction in calculation complexity: In Base-CNN model, the calculation of the convolution kernel parameters and the input matrix variables are two float-type calculations. While the calculation of BCDM is between the binary convolution kernel parameters and the float-type, that can reduce the time complexity by 60%.

c. Model deepening enhancement: After introducing the above lightweight methods, there is a certain decline in the accuracy performance, the original simple convolutional structure loses much analysis ability. Therefore, we deepen the convolutional structure, double the number of convolutional channels, and add two additional 3*3 convolutional layers. Lastly, we add max pooling layer after every two binary_convs to boost the significance of strong features and reduce over-fitting problems. After the above steps, we finish the BCDM model construction. Later, we will fully compare the volume, speed, overhead, etc. before and after compression in the experiments to verify the above lightweight improvements.

VI. Experiment

A. Setup

In this section, we mainly introduce our self-built IPv6-LAN experimental topology environment as well as the generation method of IPv6-DDoS traffic sources for our test. The experimental equipment of the IPv6-LAN is shown in Table IV.

TABLE IV: Environment for experimental hardware

Server name	Dell PowerEdge R470
Monitor host environment	OS: Ubuntu 18.04.1 CPU: Intel(R) Xeon(R)Gold 6238R 2.20GHz RAM: 16G GPU: NVIDIA GeForce GTX 1080 16G
Other host environment	OS: Ubuntu 18.04.1 CPU:Intel(R) Xeon(R)Gold 6238R 2.20GHz RAM:8G
Project environment	Python 3.7.6, Tensorflow-gpu 1.14.0, Keras 2.2.5, Cuda 10.0.130, Cudnn 7.6.5
Traffic capture tool	Wireshark 3.4.5

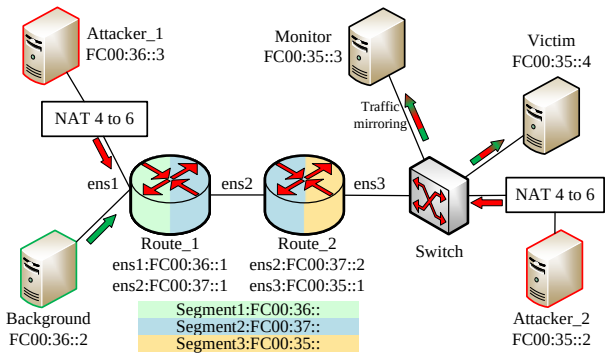


Fig. 10: IPv6-LAN: IPv6 DDoS simulation topology

Fig. 10 shows the topology of IPv6-LAN, which is constructed on the server running in the IPv6 CERNET2 environment, including 7 virtual hosts of 5 types and three network segments to simulate DDoS attack scenarios:

- Attacker_1&2: Two nodes are designated to send DDoS traffic to the victim. To address the lack of IPv6 DDoS public traffic datasets, we use NAT4to6-Jool tool on these nodes to replay the CIC-DDoS2019 dataset, generating DDoS attack traffic with IPv6 characteristics. The transparent conversion feature of NAT4to6 ensures that the converted IPv6 traffic can maintain the same communication targets and application data transmission as the original IPv4 traffic, thereby ensuring data reliability and consistency. Not only that, this experiment leverages the recognized CIC-DDoS2019 dataset as the experimental source, which can significantly improve the credibility, comparability, and repeatability of the research results, allowing other researchers to verify and compare based on the same data. The static address translation strategy of NAT4to6 is shown in the following Table V. And the replay steps includes: a. IPv4 source address rewriting: All DDoS traffic in CIC-DDoS2019 is originally sent from the source IP 172.16.0.5 to the destination IP 192.168.50.1. However, the fixed source will introduce unreasonably strong features in the model, leading to overfitting and failing to reflect the distributed nature of real DDoS attacks. To address this, we randomly rewrite the IPv4 source addresses to introduce diversity; b. Configure the NAT4to6 pool and network interface in Jool: Map the IPv4 address and network interface to the IPv6 space. Specifically, the attack node specifies FC00:35/36 as the IPv6 prefix and connects the IPv4 address in hexadecimal form; c. Set NAT4to6 rules to implement address translation: Use iptables and ip6tables to configure NAT translation rules for inbound IPv4 traffic and outbound IPv6 traffic. This setup ensures proper address translation and traffic forwarding; d. Replay CIC-DDoS2019 as IPv6 traffic: During the replay process, the intensity of the DDoS attack can be flexibly adjusted by changing the replay rate. This allows us to simulate DDoS attacks of different intensities and conduct comprehensive testing of the monitoring mechanism.

TABLE V: The static address translation of NAT 4to6

CIC-DDoS2019	IPv4	IPv6 (NAT4to6)
Source IP	***.***.***.*** For example: 172.16.0.5	FC00:36::Hexadecimal IPv4 address For example: FC00:36::AC:10:0:5 (Attacker_1) FC00:35::AC:10:0:5 (Attacker_2)
Destination IP	192.168.50.1	FC00:35::4

- Route_1&2: The soft routing node in the topology, which used to route and forward the IPv6 traffic between different network segments. CIC-DDoS2019 after NAT4to6 conversion can be correctly forwarded through routing also indirectly proves the feasibility of the replay method.
- Background: During the DDoS simulation process, it can generate normal traffic in IPv6-LAN topology as background traffic. We capture the IPv6 business traffic from CERNET2 network in advance and inject into IPv6-LAN. As the backbone of the education network, CERNET2 has a strict internal security review system; therefore, our background traffic source can be guaranteed as normal business traffic without attacks. During the background traffic replay process, we first redirect the destination to

"FC00:35::4", and then replace the source address segment with "FC00:36" to meet the routing requirements. In the experiment, the background node can cooperate with attack nodes to adjust the traffic rate, thereby simulating different DDoS intensities in the topology.
- Victim host: The target host, that used to receive DDoS attack traffic, as well as normal background traffic.
- Monitor host: Used to monitor the traffic to the victim host in real-time. By setting the mirroring command on ip6tables of Router_2, all traffic forwarding to the victim host can be mirrored and saved as pcap files.

B. Experimental dataset

In this section, based on the traffic captured from our IPv6-LAN topology, we aggregate the traffic preprocessing strategies designed in Section IV to build our traffic matrix dataset for BCDM training and testing.

1) Data sets types: As Equation 2, we define the intensity value (Int) of DDoS incident in our paper. Referencing existing work[29], it is defined as the ratio (%) of DDoS rate to the total network traffic rate. In the experiment, in order to control variables during the training process of the BCDM model to maintain the stability of learning, we regard that the traffic packets from different sources in the network are ideally uniformly distributed. Thus, the rate ratio intensity can be translated into the proportion of DDoS packets within the total number of network packets in the traffic matrix sample. We can adjust the proportion of DDoS packets in the traffic matrix to train and test the performance of BCDM under different intensities of DDoS incidents. For example, to generate a traffic matrix sample representing a network under a 10% intensity DDoS, we can randomly insert 10 DDoS attack packets into the matrix containing 100 network packets.

$$\begin{aligned} Int(\%) &= \frac{DDoS\ traffic\ rate}{Total\ network\ traffic\ rate} \\ &= \frac{No.\ of\ DDoS\ packets\ inmatrix}{100(No.\ of\ matrix\ rows)} \end{aligned} \tag{2}$$

Furthermore, we also consider two types of intensity indicators to enrich the different behaviors of DDoS attack, as shown in Table VI, including fixed-intensity and dynamic-intensity.

Fixed intensity: We set 15 traffic sets with DDoS intensities ranging from 1% to 15%, where the malicious traffic matrices in each set have same intensity.

Dynamic intensity: We set 4 traffic sets with dynamic DDoS intensity in 1%-3%, 4%-7%, 8%-11% and 12%-15% intervals, where the malicious traffic matrices of each set have random intensities within each interval, so as to simulate the intensity fluctuations among different captured traffic matrices during DDoS incident monitoring.

2) Division of samples in data set: For our traffic matrix dataset of any intensity type, it contains 10,000 traffic matrices and 10,000 labels, including 5,000 normal matrices and 5,000 malicious matrices containing DDoS packets. With a ratio of 7:2:1, We divide each data set into train set, valid set, and test set. Among them, the train set and the valid set participate in the training process to fit and tune the model parameters, whilst the test set does not participate in training process, is only used to evaluate the performance of the trained model.

C. Evaluation Metrics

In experiment, based on the confusion results during BCDM model testing in Table VII, where the number of samples with TP (positive sample predicted as positive class), TN (negative sample predicted as negative class), FP (negative sample predicted as positive class), and FN (positive sample predicted as negative class), we evaluate the performance of our BCDM model in terms of the following metrics, including: loss value during model training and testing, accuracy ($ACC = \frac{TP+TN}{TP+TN+FP+FN}$), precision($PRE = \frac{TP}{TP+FP}$), $Recall = \frac{TP}{TP+FN}$, $F1\text{ - }score = 2\frac{PRE \times Recall}{PRE+Recall}$, ROC-curve($FPR = \frac{FP}{FP+TN}$ as the abscissa, $TPR = \frac{TP}{TP+FN}$ as the ordinate), and AUC(Area Under ROC-Curve).

TABLE VII: Confusion results

True value	Predict value	
	Positive	Negative
Positive	TP	FN
Negative	FP	TN

D. Performance evaluation

The evaluation of this mechanism mainly includes five parts: the most important is RQ1. Performance evaluation of the BCDM monitoring model, including evaluation in a single intensity data set, a dynamic intensity data set, and a dynamic network; and RQ2. Performance comparison of the BCDM monitoring mechanism and related algorithms; in addition, it also includes RQ3. Efficiency evaluation of the IPv6 traffic preprocessing strategy, RQ4. Lightweight verification of BCDM, and RQ5. Feasibility analysis of our proposed IPv6-DDoS attack source replayed by NAT4to6.

1) RQ1-1: BCDM performance on traffic set: In this section, we train and test the BCDM model on the data set in Section V, and demonstrate performance of the model based on these above metrics. In training, we set train epohs = 100, batchsize = 64 and learning rate start from 0.01 with the ReduceLROnPlateau dynamical change. Table VIII shows the performance metrics of BCDM model on the train set, valid set, and test set of the fixed and dynamic intensity data

TABLE VI: Data sets types

Type	Percentage of DDoS packets in traffic matrix (DDoS intensity)														
Fixed intensity	1%	2%	3%	4%	5%	6%	7%	8%	9%	10%	11%	12%	13%	14%	15%
Dynamic intensity	1%-3%			4%-7%				8%-11%				12%-15%			

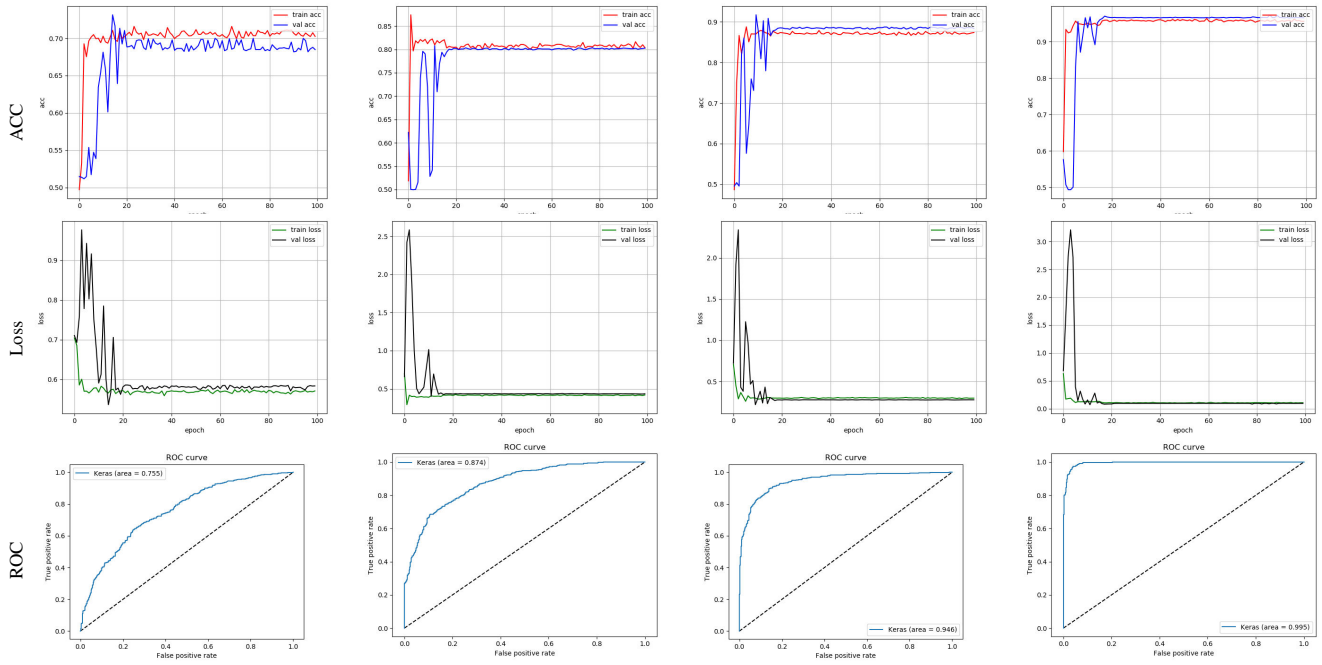
TABLE VIII: BCDM performance on IPv6-DDoS traffic matrix set

	Train set						Valid set						Test set					
Intensity	LOSS	**ACC**	F1-score	PRE	RECALL	AUC	LOSS	**ACC**	F1-score	PRE	RECALL	AUC	LOSS	**ACC**	F1-score	PRE	RECALL	AUC
1%	0.5706	**0.7024**	0.7033	0.7089	0.7047	0.7651	0.5830	**0.6850**	0.6843	0.6889	0.6866	0.7651	0.5908	**0.6920**	0.6899	0.6941	0.6904	0.7651
2%	0.4872	**0.7474**	0.7710	0.7752	0.7717	0.8405	0.4650	**0.7710**	0.7703	0.7755	0.7710	0.8405	0.4698	**0.7650**	0.7640	0.7682	0.76455	0.8405
3%	0.4194	**0.8033**	0.8047	0.8145	0.8061	0.8816	0.4340	**0.8030**	0.8017	0.8104	0.8029	0.8817	0.4202	**0.7840**	0.7792	0.7979	0.7824	0.8981
4%	0.3804	**0.8196**	0.8287	0.8308	0.8288	0.9032	0.3570	**0.8380**	0.8379	0.8396	0.8375	0.9033	0.3312	**0.8430**	0.8429	0.8446	0.8438	0.9033
5%	0.2983	**0.8734**	0.8703	0.8707	0.8703	0.9400	0.2780	**0.8830**	0.8834	0.8834	0.8835	0.9401	0.3018	**0.8720**	0.8720	0.8726	0.8723	0.9401
6%	0.2495	**0.8959**	0.8981	0.8984	0.8981	0.9493	0.2420	**0.8880**	0.8884	0.8884	0.8888	0.9494	0.2275	**0.9090**	0.9087	0.9083	0.9093	0.9494
7%	0.2004	**0.9110**	0.9145	0.9145	0.9146	0.9694	0.2070	**0.9080**	0.9078	0.9076	0.9081	0.9694	0.1904	**0.9130**	0.9127	0.9125	0.9129	0.9694
8%	0.2102	**0.9164**	0.9220	0.9220	0.9219	0.9632	0.2050	**0.9200**	0.9194	0.9194	0.9194	0.9632	0.2125	**0.9190**	0.9189	0.9189	0.9189	0.9633
9%	0.1892	**0.9254**	0.9254	0.9256	0.9254	0.9688	0.1510	**0.9440**	0.9434	0.9435	0.9437	0.9689	0.2047	**0.9180**	0.9179	0.9184	0.9180	0.9689
10%	0.1065	**0.9557**	0.9615	0.9616	0.9617	0.9873	0.0950	**0.9660**	0.9654	0.9656	0.9653	0.9873	0.0936	**0.9640**	0.9638	0.9644	0.9634	0.9873
11%	0.1065	**0.9596**	0.9687	0.9690	0.9687	0.9857	0.1000	**0.9610**	0.9609	0.9613	0.9610	0.9858	0.1018	**0.9580**	0.9579	0.9585	0.9580	0.9858
12%	0.0340	**0.9870**	0.9891	0.9891	0.9891	0.9930	0.0340	**0.9900**	0.9894	0.9896	0.9894	0.9930	0.0309	**0.9890**	0.9889	0.9890	0.9889	0.9930
13%	0.0281	**0.9873**	0.9932	0.9933	0.9932	0.9974	0.0249	**0.9915**	0.9915	0.9914	0.9915	0.9929	0.0244	**0.9920**	0.9919	0.9916	0.9924	0.9980
14%	0.0165	**0.9996**	0.9987	0.9990	0.9987	0.9957	0.0197	**0.9910**	0.9909	0.9913	0.9910	0.9858	0.0101	**0.9980**	0.9979	0.9985	0.9980	0.9970
15%	0.0181	**0.9943**	0.9994	0.9994	0.9994	0.9968	0.0024	**0.9995**	0.9995	0.9995	0.9995	0.9997	0.0049	**1.0000**	0.9990	0.9991	0.9989	1.0000
1%-3%	0.4379	**0.7866**	0.7859	0.7860	0.7859	0.8604	0.4750	**0.7680**	0.7683	0.7684	0.7682	0.8605	0.4832	**0.7700**	0.7698	0.7698	0.7705	0.8605
4%-7%	0.1586	**0.9447**	0.8721	0.8967	0.8765	0.9580	0.3060	**0.8750**	0.8744	0.8974	0.8795	0.9581	0.3046	**0.8810**	0.8785	0.9055	0.8783	0.9580
8%-11%	0.1220	**0.9460**	0.9454	0.9467	0.9459	0.9770	0.1200	**0.9470**	0.9463	0.9483	0.9458	0.9771	0.1294	**0.9420**	0.9419	0.9435	0.9420	0.9771
12%-15%	0.0145	**0.9949**	0.9967	0.9967	0.9967	0.9980	0.0090	**0.9980**	0.9975	0.9975	0.9974	0.9980	0.0112	**0.9960**	0.9959	0.9958	0.9961	0.9980

sets in IPv6. When the DDoS intensity increases from 1% to 15%, BCDM can achieve 69.2% to 100.0% ACC on the test set, while performs similarly on the train and valid sets. This demonstrates that our proposed BCDM model has good generalization ability with no overfitting, and can effectively monitor the existence of DDoS packets in the traffic matrix. In Fig. 11, we select four fixed DDoS intensity of 1%, 3%, 5% and 10%, and show the change of ACC, LOSS and ROC of BCDM training process in detail, proving that the monitoring accuracy will rise with the rise in DDoS incident intensity.

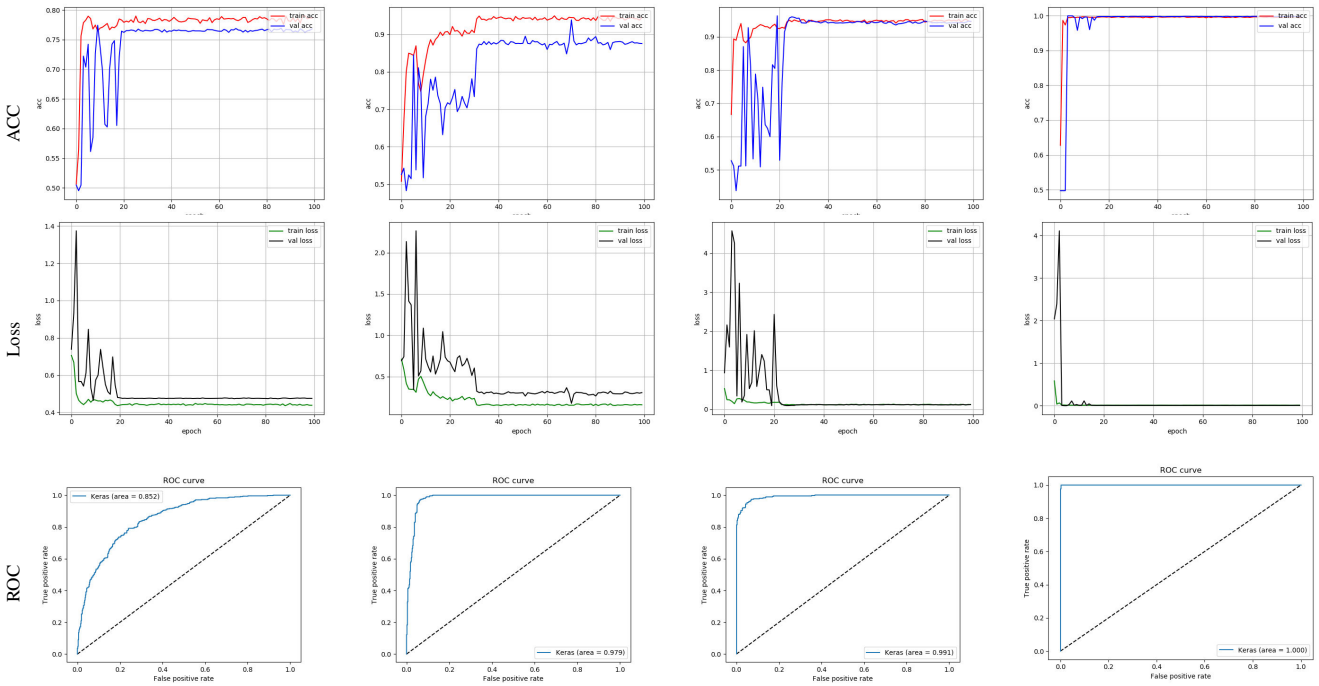
Similarly in Table VIII, BCDM can also performs well on the dynamic intensity datasets, being able to exhibit 99.6% ACC in 12%-15% intensity interval. As shown in Fig. 12, we also show the change of ACC, Loss, and ROC curve in BCDM training on dynamic intensity data sets. Overall, the performance of BCDM model on each intensity interval is roughly equivalent to the average performance on each including fixed intensity, which indicates that BCDM can still maintain performance in the face of DDoS incidents with slight fluctuations in intensity.

2) RQ1-2: Monitoring performance in dynamic network: Furthermore, we test BCDM in a real network environment, generate traffic matrices for analysis by dynamically capturing network packets. The result is shown in Fig. 13 below, where the sum of the normal background traffic and DDoS attack traffic rates is 500 packets/s. On this basis, we adjust the rates of normal traffic and DDoS attack traffic in different time intervals to simulate the network status when DDoS attacks of different intensities occur. We conduct 10 rounds of tests respectively, where the performance of BCDM in the dynamic network is consistent with that on the traffic matrix data sets. This experiment well verify the monitoring capability



(a) 1% (b) 3% (c) 5% (d) 10%

Fig. 11: Change of ACC, Loss, and ROC in BCDM training on fixed intensity data sets 1%, 3%, 5%, 10%



(a) 1%-3% (b) 4%-7% (c) 8%-11% (d) 12%-15%

Fig. 12: Change of ACC, Loss, and ROC in BCDM training on dynamic intensity data sets

of our mechanism in a realistic dynamic traffic scenario, all can realize the monitoring task at different intensities with an alarm accuracy of 95% confidence interval. But the average numbers of traffic matrices required to analyze is different. At low intensities, BCDM needs to capture multiple traffic matrices to accurately warn, but this situation will improve as the attack intensity increases. It was not until the rate of DDoS rate reach 11% that BCDM can always maintain accurate alarms through one-time analysis of the traffic matrix. The reason for this phenomenon is the instability of the captured traffic matrix samples caused by the uneven distribution of traffic in the dynamic network. It also reflects the necessity of controlling variables by assuming uniform distribution in our manually produced traffic matrix samples.

3) RQ2: Monitoring performance comparison: In the task of DDoS monitoring, the excellent solution mechanism lies in the minimum intensity of DDoS incident that it can accurately alert. The smaller this alert intensity value is, the faster the detector can react to DDoS incidents that occurs in the network. In the experiment, we use the same data set, that is, our self-built pure IPv6-DDoS traffic from the converted CIC-DDoS2019, to test the monitoring performance of different network behaviors including DDoS attack incidents of different intensities. First, based on the performance of ACC in Table IX, we draw the ACC curve of the proposed BCDM as Fig. 14, where X-axis represents the DDoS intensity, and Y-axis represents the ACC on the corresponding test set. As can be observed, BCDM can reach an ACC of more than 90.9%

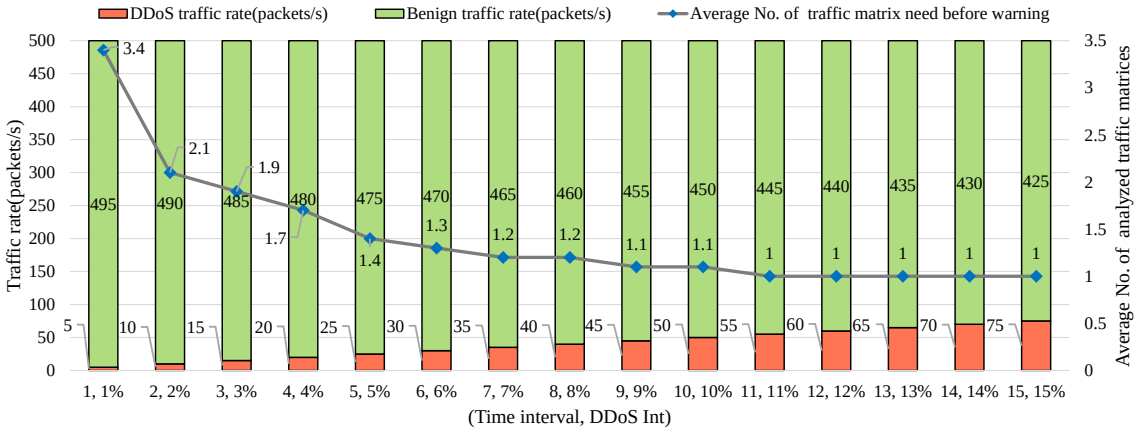


Fig. 13: Monitoring performance in dynamic network

when the DDoS intensity is over 6%, more than 95.8% when the DDoS intensity is over 10%, more than 99.2% when the DDoS intensity is over 13%, and 100% at 15% intensity.

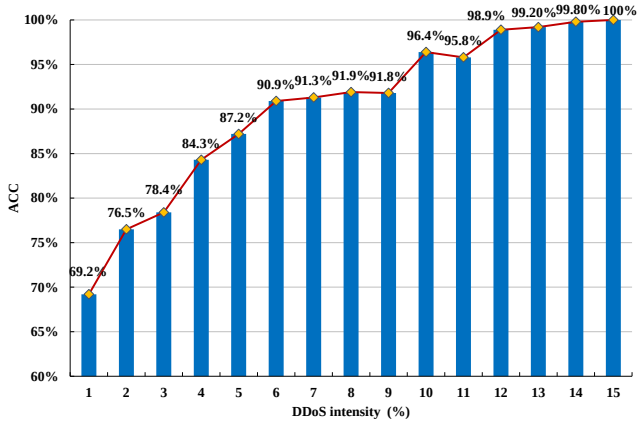


Fig. 14: BCDM DDoS monitoring sensitivity

On this basis, as shown in Table IX, we compare the recent works in Setction II.B, showing the monitoring accuracy(ACC_n) of DDoS incidents with different intensities(I_n).During this experiment, we reproduce each comparison algorithm respectively and build corresponding feature environments based on the same CIC-DDoS2019 dataset to ensure the consistency of data quality. Our comparison includes two aspects: First, as shown in Fig. 15, we test the monitoring ACC_1 and ACC_2 under 6%(I_1) and 10%(I_2) DDoS intensities to compare the accuracy performance of each method. Secondly, as shown in Fig. 16, we keep upping the intensity and record the value(I_3) when each method can reach 100%(ACC_3), so as to compare the monitoring credibility performance. In this process, the intensity span is also increased, 1 in 1%-20%, 2 in 21%-40%, and 5 over 40%.

TABLE IX: Monitoring performance comparison

Mechanism	Intensity	ACC	Intensity	ACC	Intensity	ACC
Li [29]	6%	54.80%	10%	58.00%	35%	100%
Ahalawat [30]	6%	68.81%	10%	75.53%	20%	100%
Aladaileh [31]	6%	83.12%	10%	84.97%	75%	100%
Feng [34]	6%	72.05%	10%	80.42%	22%	100%
Kirtas [35]	6%	80.29%	10%	83.65%	30%	100%
Ours	6%	**90.90%**	10%	**96.40%**	**15%**	100%

The majority of DDoS monitoring methods are based on observing the network behavior changes[27], where entropy is an important quantitative indicator of it. Such as the recent φ-entropy method Li et al.[29], the Renyi entropy method Ahalawat et al.[30], and Aladaileh et al.[31]. Generally speaking, if the entropy change exceeds 10%, the DDoS behavior can be 100% recognized, while in other cases, the accuracy can be measured based on the statistical degree of change. So although φ-entropy can reach 100% under 35% intensity, but only 54.8% and 58% accuracy at 6% and 10% intensities. Renyi entropy can perform better, reaching 100% accuracy under 20% intensity, and maintaining 68.81% and 75.53% accuracy at 6% and 10% intensities. Aladaileh's method can achieve the best 83.12% and 84.97% at 6% and 10% intensities, but cannot get close to 100% accuracy until the DDoS

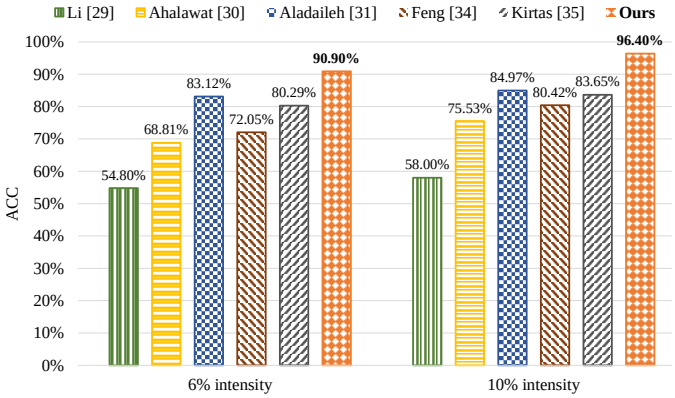


Fig. 15: Monitoring performance under 6% and 10% intensity

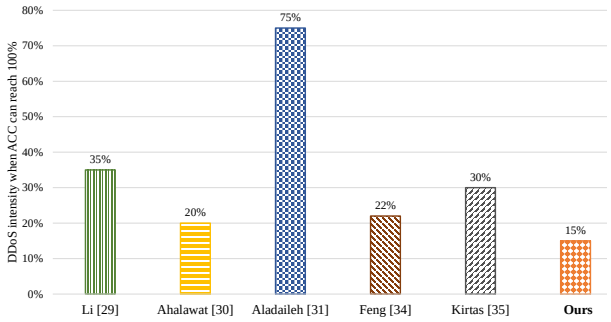


Fig. 16: The intensity of DDoS that can be 100% detected

intensity reached 75%. In addition to entropy, recent monitoring approaches also leverage techniques such as deep learning for analyzing network behavior and traffic. Feng's method[34] uses stacking of base learners to analyze flow statistics and network congestion, attaining 74.05% and 83.42% accuracies at the 6% and 10% intensities and achieves 100% accuracy at 22% intensity. Kirtas[35] introduces photonic neuromorphic deep learning and combine port information in traffic analysis to enhance monitoring, achieving 84.29% and 87.65% accuracies at the 6% and 10% intensities and hitting 100% at 30% intensity. In contrast, our proposed BCDM model can give 90.9% and 96.4% ACC at 6% and 10% intensity respectively, and 100% ACC at just 15% intensity. Obviously, our monitoring mechanism that combines BCDM and traffic matrix can more precisely monitor the emergence of the lower-intensity DDoS incidents in the network, enabling us to take defensive action more quickly.

4) RQ3: Efficiency evaluation of IPv6 preprocessing strategy: In this section, we evaluate the advantages of our proposed IPv6 traffic preprocessing strategy in Section IV.A, including its time and memory overhead. From two perspectives, first, we compare it with general algorithm library methods, such as Pyshark, to verify the efficiency advantage. Secondly, we put similar traversal idea into IPv4 traffic for comparison to verify the unique matching of our strategy to IPv6. We prepare pcap files containing 10,000 IPv6 and IPv4 packets respectively and combine the traversal strategy and pyshark to form four parsing scenarios. The time and memory overhead of pcap prasing is shown in the following Table X.

TABLE X: The time and memory overhead of pcap prasing

Scenario Name	Time taken	Memory used
Our strategy-IPv6	2.24s	24576 bytes
Pyshark-IPv6	45.74s	2777088 bytes
Traversal-IPv4	4.72s	28672 bytes
Pyshark-IPv4	33.44s	2784128 bytes

It can be seen that the parsing method of traversing the binary data packet content of pcap is extremely lightweight in terms of time and memory overhead. Compared to the existing library Pyshark[47], it is more efficient and faster because of avoiding the additional library calls and abstraction layers. Compared with IPv4, the unique fixed-length header feature of IPv6 makes our parsing strategy perform the fastest efficiency 2.24s. This is because the dynamic length of the IPv4 header and the presence of optional fields require more processing in parsing, making it inefficient and has the risk of errors, especially in high-performance network environments that need to process a large number of packets.

5) RQ4: Lightweight verification of BCDM Model: In order to meet the normalization characteristic of DDoS monitoring operation, our design of BCDM places a focus on lightweight. Therefore, we compare the running overhead of our Base-CNN model and the BCDM model to verify the effectiveness of the lightweight strategy we used. In Table XI, we compare the number of parameters, model size, memory access, FLOPs, operational intensity, reasoning time, and ACC.

TABLE XI: Lightweight of Base-CNN and BCDM

Model name	Base-CNN	BCDM
Num of pramters	113,566	11114
Model size(MBytes)	0.443	0.004
Memory access(MBytes)	1.404	0.943
FLOPs	2.2*10 7	5.3*10 7
Operational Intensity(FLOPs/Byte)	14.9	53.6
Reasoning time(ms)	0.399	0.262
(Intensity, ACC)	(10%, 98.3%)	(10%, 96.4%)

Obviously, the BCDM model has a clear advantage in lightweight and computational efficiency. First, in terms of model volume, the number of parameters and the storage size of BCDM are compressed by 90.2% and 99%. Second, BCDM includes a deeper convolutional structure to make up for the performance loss brought on by model compression. This makes BCDM have $5.3 * 10^7$ FLOPs during inference, which is 2.4 times that of Base-CNN model, while its memory access is reduced by 32.8%. Third, as Fig. 17, we draw the roof-line model[48] of the GTX1080 graphics card we used in the experiment, and indicate the performance coordinates of the two models. It can be seen that due to the difference in computing intensity, Base-CNN can only use 5.5TFLOP/s computing power, while BCDM can use full 9TFLOP/s computing power for reasoning. However, BCDM still has space for development in terms of making full use of the CPU capability even with the advancement of the graphics card. When detecting the same size traffic matrix, the reasoning time of BCDM is reduced by 34.3%. In conclusion, the lightweight BCDM can better match the requirements of DDoS monitoring scene, that means the normalized detector has a lower overhead and resource consumption.

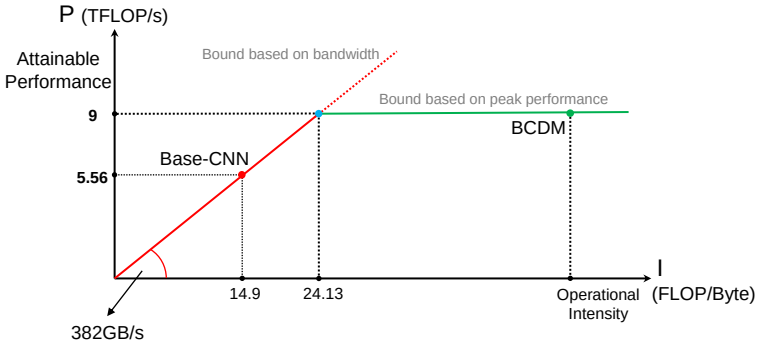


Fig. 17: Roof-line of Base-CNN and BCDM running on our GTX1080 test platform

6) RQ5: Feasibility evaluation of the NAT4to6 strategy: In this section, we introduce different DDoS attack traffic sources to verify the feasibility of the NAT4to6 strategy based on the Jool tool for replaying the CIC-DDoS2019 dataset in an IPv6 network. This involves ensuring the successful retention of DDoS attack traffic characteristics during the replay process. To achieve this, we observe the performance differences of the BCDM model under the following three ICMPv6 DDoS source as the only DDoS traffic type: a. our replayed traffic of CIC-DDoS2019; b. The ICMPv6-DDoS traffic set published by Manickam[49] with THC-IPv6 toolkit; and c. THC-IPv6 DDoS tool stimulated in Fig.10 by us.

TABLE XII: BCDM performance with different DDoS sources

DDoS traffic source	DDoS intensity	ACC
CIC-DDoS2019+NAT4to6 (ICMPv6)	5%	89.1%
CIC-DDoS2019+NAT4to6 (ICMPv6)	10%	97.7%
ICMPv6 DDoS Traffic set[49]	5%	88.5%
ICMPv6 DDoS Traffic set[49]	10%	97.0%
Thc-IPv6 in IPv6-LAN	5%	87.8%
Thc-IPv6 in IPv6-LAN	10%	95.9%

The performance is summarized in Table XII, we found that compared to traffic matrices containing ICMPv6, TCP, and UDP DDoS traffic, those containing only ICMPv6 traffic are easier to identify for malicious behavior. The reason is likely that single-protocol traffic is simpler, and the model only needs to recognize ICMPv6 characteristics without considering the complexities of TCP and UDP. Additionally, although BCDM has similar overall performance, it performs best on the NAT4to6 replay of CIC-DDoS2019, and worst on our thc-ipv6 tool, even though Manickam's dataset is also generated using thc-ipv6. This indicates that: a. Data Quality and Stability: CIC-DDoS2019, published by the Canadian Institute for Cybersecurity, features more rigorous and realistic attack simulations, making it widely recognized and used in academia and industry. b. Feasibility of NAT4to6 replay: The similar performance across the three data sources proves that the attack traffic replayed through NAT4to6 is effective, as its DDoS attack characteristics remain intact when replayed

in an IPv6 network. Although ICMPv4 and ICMPv6 have some differences in protocol details, their essential characteristics and attack patterns are the same in the context of DDoS attacks. The same is true for DDoS attacks based on TCP and UDP in IPv6 network environments. Thus, based on CIC-DDoS2019, we can introduce diversified IPv6-DDoS traffic with ICMPv6, TCP, and UDP, achieving comprehensive coverage and enhancing response of monitoring.

VII. Conclusions

The rapid expansion of access devices in the IPv6 network increases the future threat of DDoS attacks. To enhance defense capability, we need to be aware of undergoing DDoS incidents in the early stages, intervene as early as possible to reduce defense pressure. Therefore, we innovatively design a two-dimensional traffic matrix, which abstracts a network behavior feature as monitoring anchor point by aggregating continuous network traffic. On this basis, we build a monitoring core with BCDM deep learning model. After training, it can use the matrix as model input to overall perceive the malicious changes in abstracted network behavior when sporadic DDoS traffic is mixed in, thereby warning the ongoing DDoS incident. However, in this paper, the field of view for network monitoring is limited to a single traffic matrix of size 100x82, which has certain limitations. In future work, we plan to enhance traffic coverage and monitoring efficiency by expanding this field of view. Specifically, we will use the combined continuous traffic matrix samples and introduce LSTM to capture the sequence dependencies between matrices. Based on CNN's ability to analyze abstract network behavior in a single traffic matrix, the changing relationships of continuous network behaviors will be incorporated into the model analysis to explore better monitoring performance. Additionally, to promote the feasible deployment of monitoring mechanisms, we also need to comprehensively consider multiple aspects such as collaborative scalability, data privacy and security, and trustworthy supervision of monitoring activities.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No.62032013, 92267206, 62002055.

References

[1] Cybersecurity Forecast 2024, https://cloud.google.com/resources/security/cybersecurity-forecast, Last accessed 3 April 2024.

[2] Per-Country IPv6 adoption, https://www.google.com/intl/en/ipv6/statistics.htmltab=per-country-ipv6-adoption.html. Last accessed 31 March 2024.

[3] Google IPv6 Statistics, https://www.google.com.hk/intl/zh-CN/ipv6/statistics.htmltab=ipv6-adoption, Last accessed 31 March 2024.

[4] Facebook IPv6 Total Adoption, https://www.facebook.com/ipv6/tab=ipv6_total_adoption.html. Last accessed 6 April 2023.

[5] M. Tayyab, B. Belaton, and M. Anbar, "ICMPv6-based DoS and DDoS attacks detection using machine learning techniques, open challenges, and blockchain applicability: a review," IEEE Access, Vol. 8, pp. 170529-170547, 2020.

[6] 2023 DDOS THREAT INTELLIGENCE REPORT, https://www.juniper.net/content/dam/www/assets/analyst-reports/us/en/2023/corero-ddos-threat-intelligence-report. Last accessed 31 March 2024.

[7] Y. Han, L. Zhang, Y. Wang, X. Deng, Z. Gu, and X. Zhang,"Research on the Security of IPv6 Communication Based on Petri Net under IoT," Sensors, vol. 23, no. 11, 5192, 2023.

[8] H. Luo, Z. Liu, and S. Zhang, "Preventing DDoS flooding attacks with cryptographic path identifiers in future Internet," IEEE Transactions on Network and Service Management, vol. 19, no. 2, pp. 1690-1704, 2022.

[9] O. Falowo, M. Ozer, C. Li, and J. Abdo, "Evolving Malware DDoS Attacks: Decadal Longitudinal Study," IEEE Access, 2024.

[10] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection," IEEE Access, 2024.

[11] Z. Ashraf, A. Sohail, S. A. Latif, A. Hameed, and M. Y. Malik, "Challenges and mitigation strategies for transition from IPv4 network to virtualized next-generation IPv6 network," Int. Arab J. Inf. Technol., vol. 20, no. 1, pp. 78-91, 2023.

[12] K. Igulu, F. Onuodu, and T. P. Singh, "IPV6: Strengths and limitations," in Communication Technologies and Security Challenges in IoT: Present and Future, Springer, 2024, pp. 147-172.

[13] J. Zhao, X. Jing, Z. Yan, and W. Pedrycz, "Network traffic classification for data fusion: A survey," Information Fusion, vol. 72, pp. 22-47, 2021.

[14] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," arXiv preprint, no. 1602.02830, 2016.

[15] G. Lencse and N. Nagy, "Towards the scalability comparison of the Jool implementation of the 464XLAT and of the MAP-T IPv4aaS technologies," Int. J. Commun. Syst., vol. 35, no. 18, p. e5354, 2022.

[16] I. Sharafaldin, A. Lashkari, S. Hakak, and A. Ghorbaniet, "Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy," In Proc. IEEE International Carnahan Conference on Security Technology, 2019, pp. 1-8.

[17] J. Wu, J. Wang, and J. Yang, "CNGI-CERNET2: an IPv6 deployment in China," ACM SIGCOMM Computer Communication Review, vol. 41, no. 2, pp. 48-52, 2011.

[18] A. Abdallah, M. Ishak, N. Sani, I. Khan, F. Albogamy, H. Amano, and S. Mostafa, "An Optimal Framework for SDN Based on Deep Neural Network," CMC-COMPUTERS MATERIALS & CONTINUA, vol. 73, no. 1, pp. 1125-1140, 2022.

[19] A. Cil, K. Yildiz, and A. Buldu, "Detection of DDoS attacks with feed forward based deep neural network model," Expert Systems with Applications, vol. 169, no. 114520, 2021.

[20] H. Aydın, Z. Orman, and M. Aydın. "A long short-term memory (LSTM)-based distributed denial of service (DDoS) detection and defense system design in public cloud network environment," Computers & Security, vol. 118, 102725, 2022.

[21] R. Saad, M. Anbar, S. Manickam, and E. Alomari, "An intelligent icmpv6 ddos flooding-attack detection framework (v6iids) using back-propagation neural network," IETE Technical Review, vol. 33, no. 3, pp. 244-255, 2016.

[22] C. S. Shieh, T. T. Nguyen, and M. F. Horng, "Detection of unknown ddos attack using convolutional neural networks featuring geometrical metric," Mathematics, vol. 11,no. 9, 2145, 2023.

[23] S. Mahadik, P. M. Pawar, and R. Muthalagu, "Efficient intelligent intrusion detection system for heterogeneous internet of things (HetIoT)," Journal of Network and Systems Management, vol. 31, no. 1, 2, 2023.

[24] Y. Al-Dunainawi, B. R. Al-Kaseem and H. S. Al-Raweshidy, "Optimized Artificial Intelligence Model for DDoS Detection in SDN Environment," IEEE Access, vol. 11, pp. 106733-106748, 2023.

[25] A. Sharma, S. Rani, S. H. Shah, R. Sharma, F. Yu and M. M. Hassan, "An Efficient Hybrid Deep Learning Model for Denial of Service Detection in Cyber Physical Systems," IEEE Transactions on Network Science and Engineering, vol. 10, no. 5, pp. 2419-2428, 2023.

[26] A. Agarwal, M. Khari, and R. Singh, "Detection of DDOS attack using deep learning model in cloud storage application," Wireless Personal Communications, pp. 1-21, 2021.

[27] J. Yuan and K. Mills, "Monitoring the macroscopic effect of DDoS flooding attacks," IEEE Transactions on Dependable and secure computing, vol. 2, no. 4, pp. 324-335, 2005.

[28] G. Segura, S. Skaperas, A. Chorti, L. Mamatas, and C. Margi, "Denial of service attacks detection in software-defined wireless sensor networks," In Proc. IEEE International Conference on Communications Workshops, 2020, pp. 1-7.

[29] R. Li and B. Wu, "Early detection of DDoS based on φ-entropy in SDN networks," In Proc. IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference, 2020, pp.731-735.

[30] A. Ahalawat, K. Babu, A. Turuk, and S. Patel, "A low-rate DDoS detection and mitigation for SDN using Renyi Entropy with Packet Drop," Journal of Information Security and Applications, vol. 68, no. 103212, 2022.
[31] M. A. Aladaileh, M. Anbar, A. J. Hintaw, I. H. Hasbullah, A. A. Bahashwan, T. A. Al-Amiedy, and D. R. Ibrahim, "Effectiveness of an entropy-based approach for detecting low-and high-rate DDoS attacks against the SDN controller: Experimental analysis," Applied Sciences, vol. 13,no. 2, 775, 2023.
[32] Y. Xie and S. Yu, "Monitoring the application-layer DDoS attacks for popular websites," IEEE/ACM Transactions on networking, vol. 17, no. 1, pp. 15-25, 2008.
[33] B. Zhou, J. Li, Y. Ji, and M. Guizani, "Online internet traffic monitoring and DDoS attack detection using Big Data frameworks," In Proc. IEEE 14th International Wireless Communications & Mobile Computing Conference, 2018, pp. 1507-1512.
[34] Y. Feng and C. Wang, "Network anomaly early warning through generalized network temperature and deep learning," Journal of Network and Systems Management, vol. 31, no. 2, 38, 2023.
[35] M. Kirtas, N. Passalis, D. Kalavrouziotis, D. Syrivelis, P. Bakopoulos, N. Pleros, and A. Tefas, "Early Detection of DDoS Attacks using Photonic Neural Networks," 2022 IEEE 14th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), Nafplio, Greece, 2022, pp. 1-5.
[36] M. F. Saiyed and I. Al-Anbagi, "Deep Ensemble Learning with Pruning for DDoS Attack Detection in IoT Networks," IEEE Transactions on Machine Learning in Communications and Networking, 2024.
[37] Q. Li, X. Yang, Y. Wang, Y. Wu, and D. He, "Spatial–temporal traffic modeling with a fusion graph reconstructed by tensor decomposition," IEEE Transactions on Intelligent Transportation Systems, 2023.
[38] X. Wang, Z. Wang, E. Wang, and Z. Sun, "Spatial-temporal knowledge distillation for lightweight network traffic anomaly detection," Computers & Security, vol. 137, p. 103636, 2024.
[39] S. Lee, J. Ko, and S. Hong, "Facto-CNN: Memory-Efficient CNN Training with Low-rank Tensor Factorization and Lossy Tensor Compression," in Proc. Asian Conference on Machine Learning, 2024, pp. 662-677.
[40] J. He, X. Wang, Y. Song, and Q. Xiang, "A multiscale intrusion detection system based on pyramid depthwise separable convolution neural network," Neurocomputing, vol. 530, pp. 48-59, 2023.
[41] Y. Lu, M. Cai, C. Zhao, and W. Zhao, "Tor Anonymous Traffic Identification Based on Parallelizing Dilated Convolutional Network," Applied Sciences, vol. 13, no. 5, p. 3243, 2023.
[42] P.-H. C. Le and X. Li, "BinaryViT: pushing binary vision transformers towards convolutional models," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 4664-4673.
[43] A. Custura, R. Secchi, E. Boswell, and G. Fairhurst, "Is it possible to extend IPv6?," Computer Communications, vol. 214, pp. 90-99, 2024.
[44] A. R. Shaaban, E. Abd-Elwanis and M. Hussein, "DDoS attack detection and classification via Convolutional Neural Network (CNN)," 2019 Ninth International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 2019, pp. 233-238.
[45] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "CNN-Based Network Intrusion Detection against Denial-of-Service Attacks," Electronics, vol. 9, no. 916, 2020.
[46] B. Hussain, Q. Du, B. Sun and Z. Han, "Deep Learning-Based DDoS-Attack Detection for Cyber–Physical System Over 5G Network," in IEEE Transactions on Industrial Informatics, vol. 17, no. 2, pp. 860-870, 2021
[47] A. Dsouza, V. Lanjewar, A. Mahakal and S. Khachane, "Real Time Network Intrusion Detection using Machine Learning Technique," 2022 IEEE Pune Section International Conference (PuneCon), Pune, India, 2022, pp. 1-5
[48] P. Neumann and J. Kunkel, "High-Performance Techniques for Big Data Processing," In Knowledge Discovery in Big Data from Astronomy and Earth Observation. pp. 137-158, 2020.
[49] S. Manickam, A. H. B. A. AIghuraibawi, R. Abdullah, Z. A. A. Alyasseri, K. H. Abdulkareem, M. A. Mohammed, and A. Alani, "Labelled dataset on distributed denial-of-service (DDoS) attacks based on Internet Control Message Protocol Version 6 (ICMPv6)," Wireless Commun. Mobile Comput., vol. 2022, no. 1, p. 8060333, 2022.

Yufu Wang received the B.S. and M.S. degree in computer science from the Northeastern University, Shenyang, China in 2017 and 2019. He is currently working toward the Ph.D. degree in Computer Application Technology from the Northeastern University, Shenyang, China. His research interests include cybersecurity, DDoS attack defense, deep Learning, and etc.

Xingwei Wang received the B.S., M.S., and Ph.D. degrees in computer science from the Northeastern University, Shenyang, China in 1989, 1992, and 1998 respectively. He is currently a Professor at the College of Computer Science and Engineering, Northeastern University, Shenyang, China. His research interests include cloud computing and future Internet, etc. He has published more than 300 journal articles, books and book chapters, and refereed conference papers. He has received several best paper awards.

Qiang Ni is currently a Professor and the Head of the Communication Systems Group, School of Computing and Communications, Lancaster University, U.K. His research interests include the areas of future generation communications and networking, including green communications and networking, millimeter-wave wireless communications, cognitive radio network systems, non-orthogonal multiple access (NOMA), 5G and 6G, IoTs, cyber physical systems, AI and machine learning, and vehicular networks. He has authored or co-authored 300+ papers in these areas. He was an IEEE 802.11 Wireless Standard Working Group Voting Member and a contributor to various IEEE wireless standards.

Wenjuan Yu received her PhD degree in Communication Systems from Lancaster University, UK. She is currently a Lecturer with the School of Computing and Communications (SCC), Lancaster University. She was a Research Fellow with the 5G Innovation Centre (5GIC), Institute for Communication Systems, University of Surrey, UK, from 2018 to 2020. Prior to that, she worked as a part-time Research Officer at the School of Computer Science and Electronic Engineering, University of Essex, UK, from Aug. 2017 to Jan. 2018. Her research interests include radio resource management, low latency communications, and machine learning for communications.

Min Huang received the B.S. degree in automatic instrument, the M.S. degree in systems engineering, and Ph.D. degree in control theory from the Northeastern University, Shenyang, China in 1990, 1993, and 1999 respectively. She is currently a Professor at the College of Information Science and Engineering, Northeastern University, Shenyang, China. Her research interests include modeling and optimization for logistics and supply chain system, etc. She has published more than 100 journal articles, books, and refereed conference papers.