Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.039280
Article

A Modified PointNet-Based DDoS Attack Classification and Segmentation in Blockchain

Jieren Cheng[1,3], Xiulai Li[1,2,3,4,*], Xinbing Xu[2,3], Xiangyan Tang[1,3] and Victor S. Sheng[5]

[1]School of Computer Science and Technology, Hainan University, Haikou, 570228, China
[2]School of Cyberspace Security, Hainan University, Haikou, 570228, China
[3]Hainan Blockchain Technology Engineering Research Center, Hainan University, Haikou, 570228, China
[4]Hainan Hairui Zhong Chuang Technol. Co. Ltd., Haikou, 570228, China
[5]Department of Computer Science, Texas Tech University, TX, 79409, USA
*Corresponding Author: Xiulai Li. Email: lixiulai01@hainanu.edu.cn
Received: 20 January 2023; Accepted: 13 April 2023; Published: 26 May 2023

Abstract: With the rapid development of blockchain technology, the number of distributed applications continues to increase, so ensuring the security of the network has become particularly important. However, due to its decentralized, decentralized nature, blockchain networks are vulnerable to distributed denial-of-service (DDoS) attacks, which can lead to service stops, causing serious economic losses and social impacts. The research questions in this paper mainly include two aspects: first, the classification of DDoS, which refers to detecting whether blockchain nodes are suffering DDoS attacks, that is, detecting the data of nodes in parallel; The second is the problem of DDoS segmentation, that is, multiple pieces of data that appear at the same time are determined which type of DDoS attack they belong to. In order to solve these problems, this paper proposes a modified PointNet (M-PointNet) for the classification and type segmentation of DDoS attacks. A dataset containing multiple DDoS attack types was constructed using the CIC-DDoS2019 dataset, and trained, validated, and tested accordingly. The results show that the proposed DDoS attack classification method has high performance and can be used for the actual blockchain security maintenance process. The accuracy rate of classification tasks reached 99.65%, and the accuracy of type segmentation tasks reached 85.47%. Therefore, the method proposed in this paper has high application value in detecting the classification and segmentation of DDoS attacks.

Keywords: Blockchain; DDoS; PointNet; classification and segmentation

1 Introduction

A mesh topology is used to build the peer-to-peer (P2P) network structure that makes up the blockchain [1]. In the context of a blockchain network, each node can simultaneously serve as a client and a server. The blockchain network can offer consumers more dependable, high-quality, and secure

services while also more effectively distributing network traffic and offering them a wider range of service options [2]. Since every node on the blockchain network can send and receive data, traffic won't be centralized, as it is in client/server networks, but distributed over all of the network nodes instead [3]. Due to the distributed network architecture used by the blockchain system, several connection points may make network nodes more vulnerable [4]. Additionally, because the blockchain system offers a public database, attackers can quickly get all system-related data to attack the system [5].

One of the most significant risks to blockchain security is the DDoS attack [6], which is a network attack in which attackers attempt to overwhelm a network or server with traffic from various sources to disrupt service [7]. DDoS attacks have different characteristics in the blockchain ecosystem than they do in a typical network context [8,9]. In contrast to a DDoS attack in a traditional network environment, the attacker can use numerous attacked nodes to start the attack, overwhelming the target node's inbound connection and ultimately causing the network to go down [10]. Additionally, attackers may use network congestion as a means of interfering with the network's regular operations and lowering the availability and dependability of nodes [11].

DDoS attacks frequently exhibit concurrency in the blockchain ecosystem, where several and various attack types may manifest simultaneously [12]. Blockchain technology is a decentralized system in which every node is equal and can participate in the consensus process of the network. However, this also means that any node can be targeted by attackers, who can exploit their vulnerabilities to compromise the entire network [13]. Therefore, it is crucial to protect blockchain applications from DDoS attacks [14]. Studying DDoS attacks can help blockchain application developers and security experts develop more effective defense strategies, such as mitigating DDoS attacks with distributed defense mechanisms, or monitoring and identification techniques to detect and respond to DDoS attacks early [15,16]. In addition, studying DDoS attacks can also help blockchain application developers and security experts better understand network protocols and system vulnerabilities and take steps to mitigate them [17,18]. Different forms of attacks, including LDAP (Lightweight Directory Access Protocol), MSSQL (Microsoft SQL Server), NetBIOS (NetBIOS Services Protocols), Portmap, Syn (Synchronize Sequence Numbers), UDP (User Datagram Protocol), UDPLag, etc., cannot be properly detected by current DDoS attack detection techniques. Therefore, it is very important to detect DDoS attacks in a blockchain system, where a blockchain node may face multiple data streams at the same time. It is necessary to carry out parallel and coordinated detection of the data flows faced, and the detection mainly involves two aspects: the first is to detect whether there is a classification of DDoS attacks in multiple data flows at the same time, and the second aspect is to segment different data entries into DDoS attack types when DDoS attacks are detected. We construct a targeted dataset for the above problems, simulating multiple data streams that may occur at the same time, and detecting DDOS attacks based on M-PointNet networks.

Many scholars have studied and analyzed DDoS attacks in blockchain and given different ways to solve this problem. Artificial intelligence algorithms have developed into one of the workable options for identifying DDoS attacks [19,20]. The proposed framework is characterized by a high accuracy rate in detecting emerging DDoS attacks and its lightweight algorithm [21]. In [22], authors combined and took advantage of both machine learning algorithms and the Bloom filters. Kasim detected DDoS assaults with AE and SVM. The approach was 99.41% accurate on CICIDS(Canadian Institute for Cybersecurity Intrusion Detection Systems) and 99.5% on NSL-KDD [23]. Gopal and Virender introduced voting extreme machine learning (ELM) (V-ELM) to detect DDoS attacks in cloud computing [24], And it achieved 99.18% with the NSL-KDD dataset and 92.11% with the ISCX dataset. In [25], it is suggested to examine cloud provider income packets to detect and avoid DDoS TCP flood attacks. As for the datasets, [26] is the first to use the CICDDoS-2019 dataset,

which contains 12 attack types. They used multiple denoising, tensor decomposition, and classifiers to detect assault and reported binary classification accuracy >99% for various denoising algorithms. These studies do not address multi-class classification, which security professionals need to detect DDoS attack types. Aamir et al. developed a clustering-machine learning method employing network flow traffic data as feature vectors. Their technique was 96.66% accurate on their dataset and 82% on CICIDS-2017 [27]. Kachavimath et al. extracted 8 features from 41 in the DSL-KDD dataset using co-relation-based feature selection. KNN had 98.51% accuracy and Naive Bayes 91.31% [28]. In [29], The captured traffic is processed to fetch its various features, and machine learning is applied for classification that can distinguish the attack traffic from the regular traffic. The results mentioned in the text are organized as follows in Table 1:

Table 1: A breif summary for the results related in this paper

Authors	Description	Results
Tseung et al.	Using machine learning algorithms and the Bloom filters	Not mentioned
Kasim	Using AE and SVM	99.41% on CICID and 99.5% on NSL-KDD
Kushwah et al.	Voting extreme machine learning	99.18% on NSL-KDD and 92.11% on ISCX
Maranhão et al.	Multiple denoising, tensor decomposition	>99%
Aamir et al.	Develop a clustering-machine learning method	96.66% on their dataset and 82% on CICIDS-2017
Kachavimath et al.	Using co-relation-based feature selection.	KNN had 98.51% accuracy and Naive Bayes 91.31%

In this paper, we investigated a range of typical and aberrant patterns using deep learning in this work using the CIC-DDoS 2019 dataset. First, we processed and screened data using statistical techniques, and we utilized the traditional decision tree method to screen features. Then, we ultimately categorize whether there is a DDoS attack and what kind of attack type it is, using the modified PointNet network. This study offers a parallel, accurate, and effective detection approach for the blockchain environment, which can contribute to the security assurance of the contemporary blockchain system, as demonstrated by trials.

2 Background

2.1 Blockchain

As a distributed recording system, blockchain technology enables transactions to be validated and documented without the need for a single administrator. To ensure that they are securely shared among several computers, it is made up of blocks, which are collections of transaction records. Although it was initially created for bitcoin, blockchain technology is now widely used in a wide range of industries, including financial services, logistics and supply chain management, public administration, and identity and access management [30].

Typically, a blockchain system consists of the following parts:

(1) Block: To avoid tampering, a block is a collection of transaction data that have been encrypted and hashed together.
(2) A blockchain is an ordered list of blocks, where each block has a hash value that points to the block before it.
(3) Node: A blockchain network's nodes are computers. New transaction records can be accepted, verified, and stored by them.
(4) Consensus mechanism: A new block's acceptance into the blockchain is decided by using the consensus process. Nodes submit new blocks by using a proof-of-work mechanism, which requires them to solve a challenging computational issue. This stops rogue nodes from sending copious amounts of useless data or tampering with the transaction records already in place. As it guarantees the security and dependability of the blockchain network, the consensus process is a crucial part of blockchain technology. There are several distinct consensus processes used by blockchain networks, including proof-of-work, proof-of-stake, and consensus rotation.

Blockchain technology is perfect for a wide range of applications, including financial services, logistics and supply chain management, government and public services, identity identification, and access control due to its decentralization and high level of security. Blockchain technology is still evolving, but it has already gained widespread adoption and is anticipated to have a bigger impact in the future.

As shown in Fig. 1, we assume that this blockchain network consists of several nodes, which can be divided into two categories: normal nodes and nodes attacked by DDoS. Normal nodes are nodes that are running normally. They are responsible for receiving and processing transaction requests from other nodes and can update the state of the blockchain. Nodes attacked by DDoS attacks may not work properly, and may not be able to receive and process transaction requests from other nodes. In this blockchain scenario, normal nodes and nodes attacked by DDoS may exist in the same network, and both can communicate with other nodes. However, since nodes attacked by DDoS may not work properly, they may not be able to process transaction requests from other nodes on time. This may lead to a reduction in the efficiency of the entire network and may affect the stability of the blockchain.

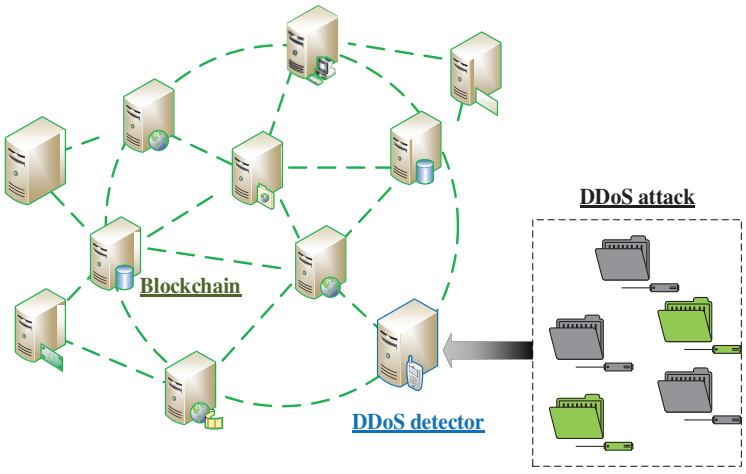

Figure 1: DDoS attacks in blockchain

2.2 CIC-DDoS2019 Dataset

A collection of datasets for detecting DDoS attacks is called the CIC-DDoS2019 dataset. Northwestern University and the Canadian Telecommunications Research Institute collaborated to build it. It can be used to train and test deep learning models to detect DDOS assaults because it comprises legitimate and malicious traffic data from actual networks. First, with 150 million packets in total, the CIC-DDoS2019 dataset is incredibly enormous. The massive amount of data needs to learn complicated patterns, which makes it perfect for deep learning model training. Second, in addition to regular traffic from numerous protocols, the CIC-DDoS2019 dataset also includes five different kinds of DDoS attacks. Due to its high diversity, it can more accurately represent actual situations. Accordingly, the model developed utilizing the CIC-DDoS2019 dataset might be more versatile and more equipped to identify various DDoS attacks. The CIC-DDoS2019 dataset also contains some extra data, such as the attack's length and target. Researchers can use this information to describe attacks and evaluate their effects. Additionally, by using this data to train models, it is possible to identify malicious traffic. Additionally, because the CIC-DDoS2019 dataset contains real data, the trained model may be more applicable to a wider range of real-world scenarios.

2.3 DDoS Attack Categories

In this study, we used the CIC-DDoS2019 dataset to undertake deep learning-based research on a variety of normal and pathological patterns, including BENIGN, LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag [31]. The first screening and processing of the feature quantity are followed by the selection of the 14 most crucial features using the traditional decision tree scheme, and then the PointNet network is utilized to categorize whether it is DDoS and what type of attack it is.

We primarily identified the following seven types of assaults in this paper.

(1) LDAP: For requesting and changing directory information within a computer network, LDAP is a widely used network protocol. The term 'LDAP type attack' describes a DDoS attack in which the attacker floods the target network's server with numerous forwarded LDAP queries, preventing it from responding to legitimate requests and achieving the denial of service goal.
(2) MSSQL: Microsoft's MSSQL is a database management system that controls databases using the SQL language. In a DDoS assault, an 'MSSQL type attack' is when the attacker floods the target network's server with a lot of forged MSSQL requests, preventing it from responding to legitimate requests and achieving the denial of service goal.
(3) NetBIOS: The local area network protocol known as NetBIOS allows machines on the network to exchange files and printers. In DDoS assaults, NetBIOS attacks are when the attackers flood the target network's server with a high number of fraudulent NetBIOS requests, preventing it from responding to legitimate requests and achieving the denial of service goal.
(4) Portmap: A application called Portmap is used in Linux systems to manage the mapping between ports and services. In DDoS assaults, the term "Portmap type attack" describes how the attacker floods the target network's server with a high number of fraudulent Portmap requests, preventing it from responding to regular requests and achieving the denial of service goal.
(5) Syn: A data packet called a Syn is used to start a TCP connection. In typical TCP communication, the client sends a Syn packet to initiate a connection request, the server responds with a Syn-ACK packet to confirm the connection, and the client then sends a second ACK message to reiterate connection confirmation. In contrast, a Syn attack involves the attacker forging a lot of Syn packets, which forces the server to deal with an excessive amount of connection requests and serves the denial-of-service goal.

(6) UDP: UDP is a connectionless transport layer protocol that allows for the transmission of data over a network, but it does not provide secure data delivery. The goal of a UDP-type attack is to deny service by forcing the server to process an excessive amount of data packets by forging a large number of UDP data packets.
(7) UDPLag: UDPLag-type attacks can place a heavier demand on the server than UDP-type assaults because they aim to increase the server's network connections. In a UDPLag-type attack, the attacker will create a large number of forged UDP packets, each of which will contain randomly generated source and destination port numbers. This will force the server to create a lot of network connections to process the packets, which will serve the attacker's goal of denial of service.

3 DDoS Attack Classification and Segmentation Based on PointNet

3.1 Method Structure

In order to realize the classification and segmentation of DDoS attacks for blockchain systems described above, we propose the following algorithm processing flow in this section, and the main method structure is described as follows:

(1) Data Pre-process: preliminary processing of the original CIC-DDoS2019 dataset, random samples and preliminary screening of features.
(2) Feature screening: The decision tree method is used to screen the processed data set based on key features.
(3) Dataset Generating: Build corresponding specialized datasets according to the needs of classification and segmentation.
(4) M-PointNet-based Modeling: The M-PointNet network is designed for the classification of DDoS attacks and the segmentation of various DDoS attack types, and the structure of the network is discussed.
(5) Model training and applying: The proposed model is trained based on the dataset and applied to the test of DDoS attack detection.

3.2 Data Pre-Process

To compare each attack mode fairly in this section, we first create a dataset. To achieve this, we randomly select a subset of DDoS attack entries from the original CIC-DDoS2019 dataset. In the dataset we created, the percentages of entries with and without DDoS attacks are both 50%, as are the percentages of entries using the seven distinct attack tactics.

As shown in Fig. 2, 10,500 pieces of data with BENIGN tags and 1,500 pieces with each of the following labels: LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag.

Except for the tag column, there are 87 feature quantities in the read CIC-DDoS2019 dataset. The first thing we noticed was that 12 feature amounts, such as Bwd PSH Flags, Fwd URG Flags, and PSH Flag Count, were constant throughout. We will remove the DDoS detection because it is ineffective and leave 75 remaining functionalities. Then, we exclude several feature quantities, including six, such as Flow ID, Source IP, Destination IP, Timestamp, SimilarHTTP, Unnamed:0, etc., that do not have the usual statistical relevance, leaving a total of 69 feature quantities. The correlation between the aforementioned remaining feature quantities was then counted; Fig. 3 displays the correlation. A few feature characteristics that we found to be strongly associated include the Total Length of Bwd Packets and Fwd IAT greater than 0.9. A total of 39 feature quantities remain after the deletion of 30 feature

quantities, including Total, Bwd IAT Total, etc. There are now just 29 feature quantities left after the removal of the features Fwd PSH Flags, Syn Flag Count, CWE Flag Count, Active Mean, Active Std, Active Max, Active Min, and Idle Std with additional 0 values.

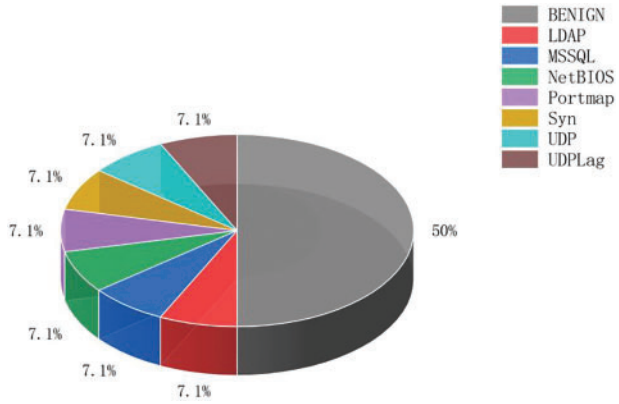

Figure 2: Proportion of DDoS attack types in the constructed dataset

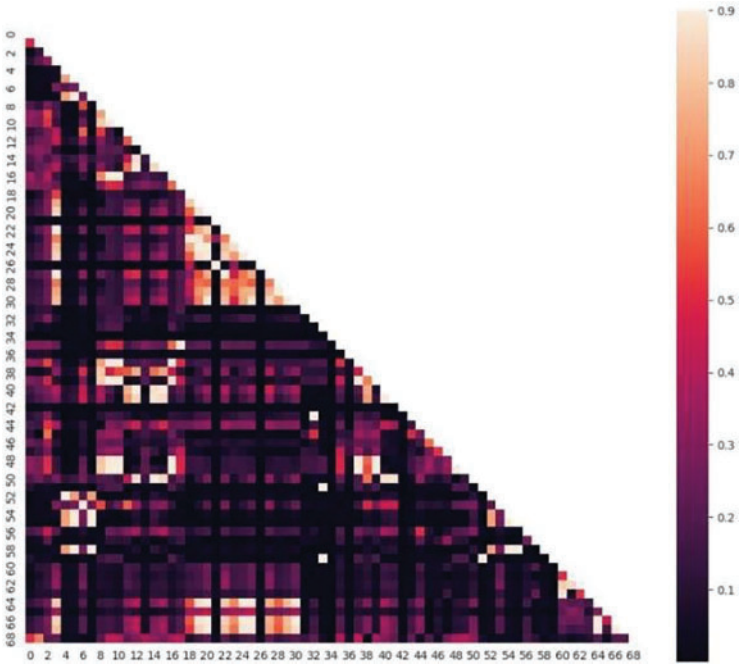

Figure 3: The correlation between the 69 features

3.3 Feature Screening

The features are then further screened using a straightforward decision tree method. A decision tree is a type of tree structure used to categorize and forecast data [32]. To divide the dataset into subsets and partition the features, decision trees require training data. Each leaf node represents a category, each branch reflects the value of each internal node's representation of a feature.

The fundamental ideas and procedures in decision tree classification [33]:

(1) Data preparation: You must first gather training datasets, which include input features and associated output categories.
(2) Choose the best feature: To choose the best feature for division, consider information gain or the Gini index. The Gini index is chosen as the index in the training for this article. The Gini index of the k category distribution is as follows, the larger the Gini index, the greater the uncertainty of the sample. K represents the value that has the most possible variety of cases in a discrete probability distribution.

$$\text{Gini}(D) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2 \tag{1}$$

(3) Create a decision tree: Create a decision tree using the features that have been chosen. The dataset is split into two subsets—one for each feature and one without—and a decision tree is recursively constructed for each subset.
(4) Decision tree pruning: To avoid overfitting, the decision tree is pruned once it has been built.
(5) Decision tree classification: input data samples and categorize using a decision tree. Beginning at the root node, it proceeds to the leaf node by searching through each layer by the feature value of the input sample. The classification outcome of the input sample is the category that corresponds to the leaf node.

We perform training based on the fundamental decision tree using the aforementioned standardized dataset, and the training results are displayed in Fig. 4 as a result. The interior nodes of the tree structure's characteristics are among its most significant elements. We group them according to importance, and the total includes the following features: Source Port, Destination Port, Protocol, Total Forward Packets, Total Backward Packets, Forward Packet Length, Backward Packet Length Max, Flow Bytes/s, Flow IAT Mean, Flow IAT Std, Max Packet Length, ACK Flag Count, URG Flag Count, Inbound, and another 14 features.

3.4 Dataset Generating

In the subsequent training, we filter the data for the crucial features Source Port and Destination Port, keeping just 10 of the port numbers that are commonly used while setting the values of the remaining port numbers to −1. Then, to create a set of standardized data, we normalize each feature quantity by taking its value and subtracting it from its mean value, and dividing it by its standard deviation.

Here is how we simulate the DDoS detection procedure. Since many feature entries (one row of data) can be gathered concurrently, the deep learning-based detector design has the following two issues: Detecting the current multiple is one. The first is to determine whether a DDoS attack exists in the entry, which is a classification issue, and the second is to categorize the various DDoS attack kinds individually when determining whether there are DDoS attacks in the current numerous entries, which is a division of DDoS attack types.

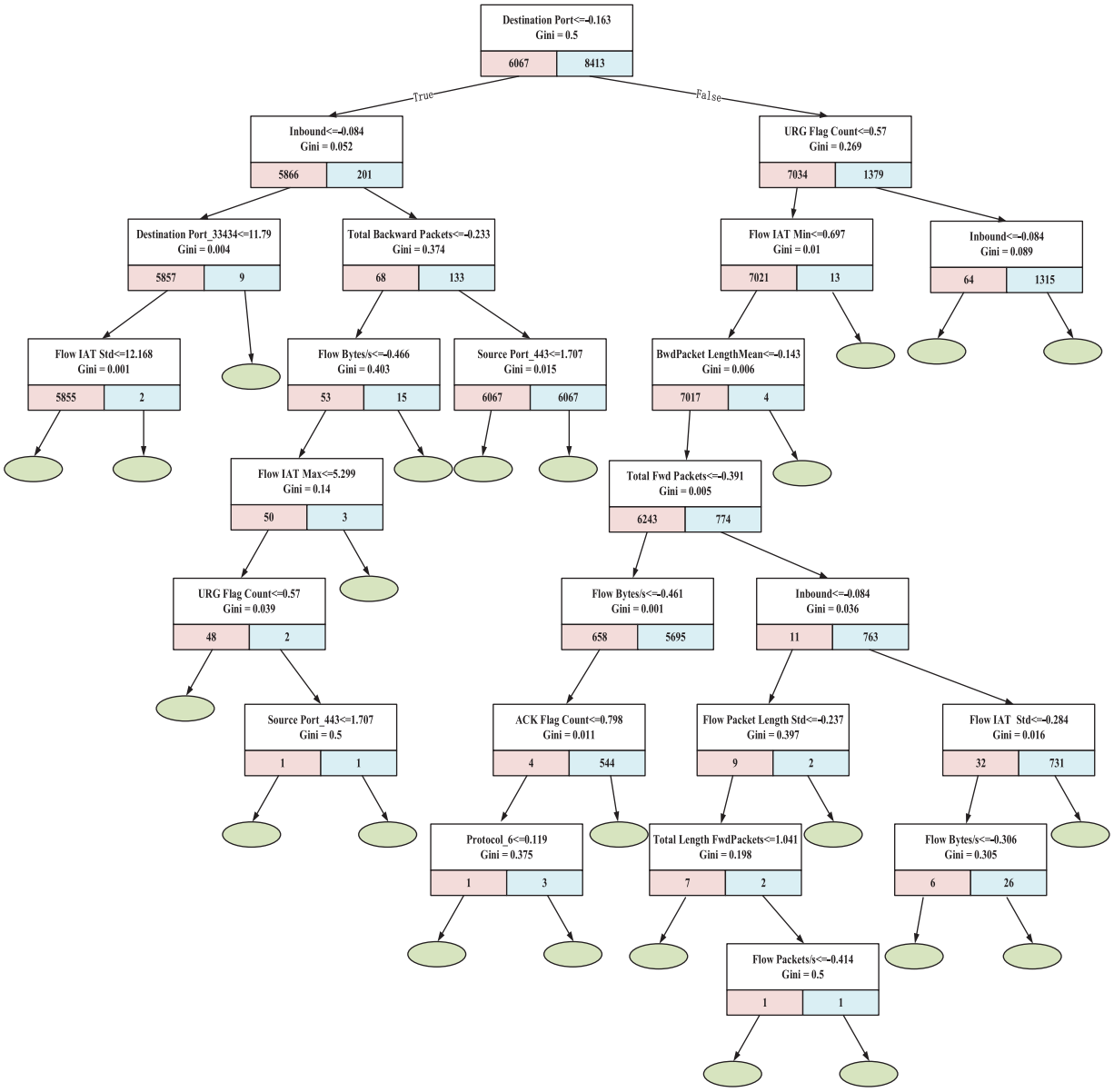

Figure 4: The results using decision tree for feature screening

The next dataset, which may be utilized for real-time DDoS detection, was created using random extraction based on the 14 significant features that were screened above, as shown in Table 2. No DDoS attack category is represented by category '0', while a DDoS attack category is represented by category '1'. Fig. 5 displays the data value and label value of random samples. Each element in each sample can be shown to have a one-to-one correlation with its label value, and the numbers 1 through 8 stand for the BENIGN, LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag types, respectively. It is noted that in samples with DDoS attacks, the number and types of DDoS attacks that may occur simultaneously may be different.

Table 2: An example of the dataset used in this work

Attribute	1	2	3	4	5	6	7	8	9	10	11	12	13	14	label
item1	−0.35	−0.04	1.12	0.01	0.01	−0.51	4.05	−0.47	−0.24	−0.28	3.00	−2.06	0.58	1.09	**1**
item2	−0.35	−0.04	1.12	−0.04	−0.01	−0.49	−0.25	−0.47	−0.22	−0.27	−0.53	0.48	−1.72	1.09	**1**
item3	−0.35	−0.38	−0.90	−0.03	−0.02	−0.51	−0.10	−0.47	−0.23	−0.28	−0.41	0.48	0.58	1.09	**1**
item4	−0.35	−0.04	1.12	−0.03	−0.03	−0.51	−0.26	−0.47	0.39	0.25	−0.54	−2.06	0.58	1.09	**1**
item5	2.94	−0.38	−0.90	−0.03	−0.03	−0.51	−0.26	1.03	−0.24	−0.28	−0.54	0.48	0.58	−0.91	**3**
item6	−0.35	−0.38	−0.90	−0.03	−0.03	3.21	−0.26	3.82	−0.24	−0.28	1.76	0.48	0.58	−0.91	**2**
item7	−0.35	−0.38	1.12	−0.03	−0.02	−0.49	−0.25	−0.47	−0.24	−0.28	−0.53	−2.06	0.58	−0.91	**5**
item8	−0.35	−0.38	−0.90	−0.03	−0.03	0.07	−0.26	0.20	−0.24	−0.28	−0.18	0.48	0.58	−0.91	**4**

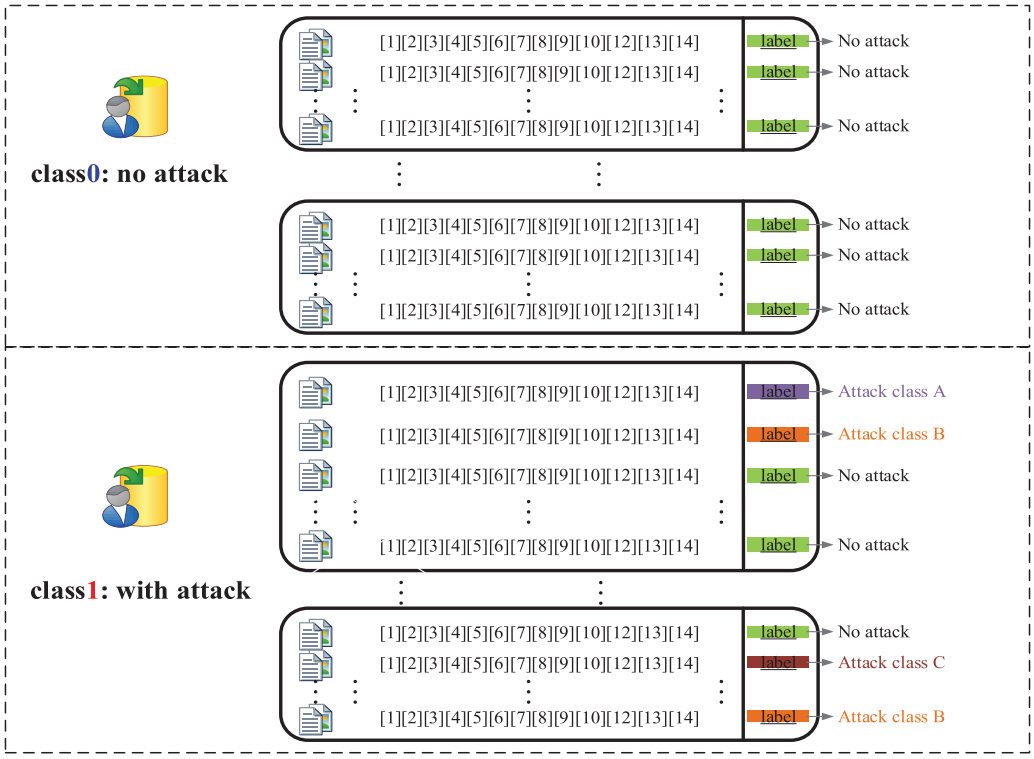

Figure 5: Dataset diagram with values and labels

3.5 M-PointNet-Based Modeling

A deep learning architecture called PointNet is used to process point cloud data. It uses a multi-layer perceptron to learn point cloud features and has a symmetric architecture to process point clouds quickly (MLP). Therefore, DDoS attacks can be detected using the PointNet network [34]. To discover probable attack patterns and efficiently learn network traffic features, the PointNet network can analyze network traffic to detect DDoS attacks. It can be trained to distinguish between legitimate and malicious traffic and can classify fresh incoming traffic. PointNet processes point cloud data more efficiently than other computers because it uses global shared feature extraction technology, which can

handle a variety of point clouds and is not order-sensitive. The quantity of point clouds frequently places restrictions on learning methods [35].

We start by enhancing the PointNet network in this part. Processing the 3D point cloud makes use of the standard PointNet network, which has three channels in all. We suggest a 14-channel PointNet to address the issue of DDoS detection, and Fig. 6 depicts the designed PointNet's network layout.

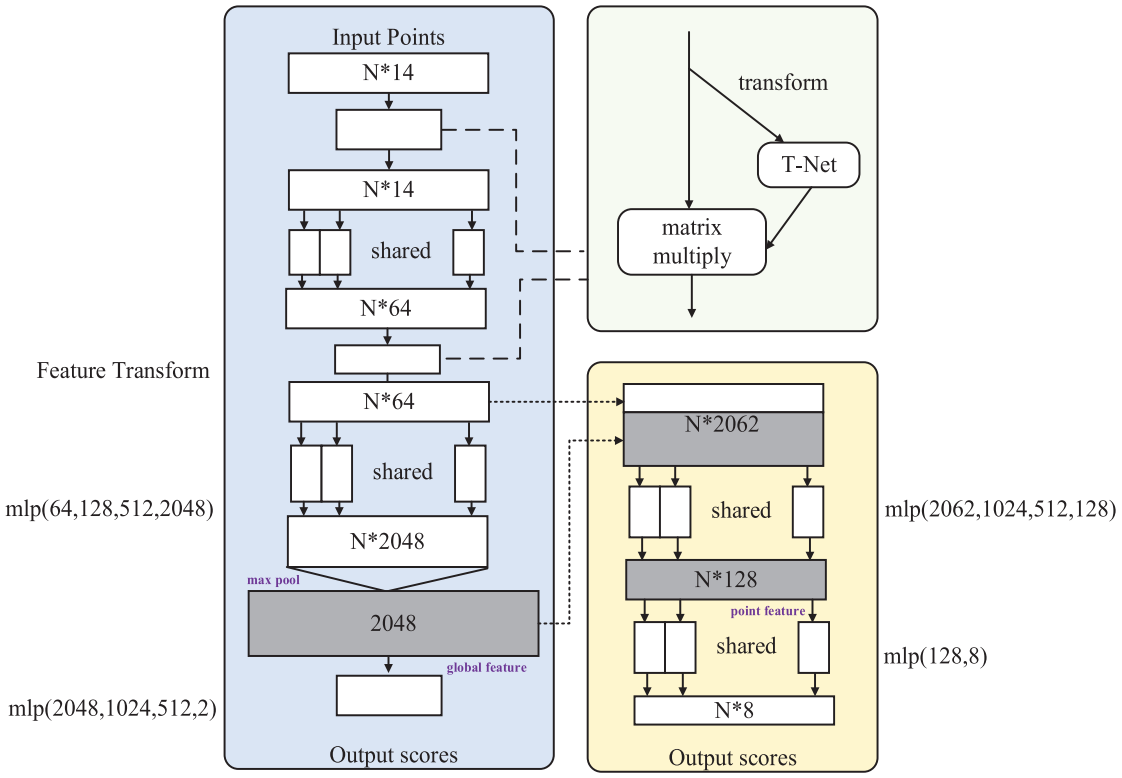

Figure 6: A layout of M-PointNet proposed in this work

The modified PointNet network structure is as follows: The network accepts a data sample with several entries, each point having 14 feature quantities, in its input layer. The global branch and local branch of the network then translate low-dimensional features to high-dimensional space through the feature dimension enhancement operation, making it simpler for the network to learn complicated feature representations and enhancing classification and segmentation accuracy. In the global branch, global features are accomplished by mapping low-dimensional features into a high-dimensional space using a fully-connected layer. In the local branch, local features are accomplished by mapping low-dimensional characteristics onto a high-dimensional space using a point convolutional layer. The global features obtained through max-pool have a total of 2048 elements and are used in the detection of the classification problem of whether there is a DDoS attack. In the local branch, we concatenate the 64-dimensional high-dimensional features obtained by feature transformation with the 2048-dimensional global features to obtain a 2062-dimensional feature tensor. Therefore, the segmentation result of the DDoS attack type can be obtained after processing by the fully connected layer.

The following is the point convolution computation process:

Let the input point cloud be $X \in \mathbb{R}^{N\times C}$, which N represents the number of points in the point cloud, and C represents the number of features of each point. The output of the point convolution layer is $Y \in \mathbb{R}^{N\times D}$, which D represents the number of features output. The parameters of the point convolution layer are $W \in \mathbb{R}^{C\times D}$. The calculation process of the point convolution layer is as $Y = XW$.

Fewer parameters are needed to extract the point cloud's global features in the classification and segmentation networks as a result of the dimensionality reduction mapping process, which converts high-dimensional point cloud data to low-dimensional space. This enhances the networks' capacity for generalization. This layer concatenates the outputs of the global branch and the local branch at the output layer of the network and uses a fully connected layer to transfer them to the output space. The output layer typically uses a softmax activation function for classification tasks and a sigmoid activation function for segmentation tasks. Let the input be, where is the number of classification categories.

The softmax function is calculated in the manner described below [36]:

$$softmax\,(z_i) = \frac{\exp\,(z_i)}{\sum_{j=1}^{K} \exp\,(z_j)} \tag{2}$$

The sigmoid function is calculated in the manner described below:

$$\text{sigmoid}\,(z) = \frac{1}{1+\exp\,(-z)} \tag{3}$$

The sigmoid function produces a number between 0 and 1, which can be used to express the likelihood that a sample belongs to a particular class.

There are two processes involved in DDoS detection. The first step is to identify and categorize any DDoS attacks. When there is no DDoS attack, we use the label 0; when there is a DDoS assault, we use the label 1, and we use the cross-entropy loss function as the optimizer:

$$\text{CE}\,(y,t) = -\sum_{i=1}^{K} t_i \log y_i \tag{4}$$

Second, we select the quadratic square error loss function (MSE loss function) as the loss function for the type segmentation task under DDoS:

$$\text{MSE}\,(y,t) = \frac{1}{K}\sum_{i=1}^{K} (y_i - t_i)^2 \tag{5}$$

Adam is chosen as the optimizer to iteratively change the network's backpropagation parameters throughout the training phase [37].

4 Simulation and Analysis

In this section, using the enhanced PointNet network previously suggested, we simulate the DDoS assault classification and segmentation method. We train and deploy the model according to the following process:

(1) Build a dataset of DDoS attacks in the blockchain system shown in Fig. 5 and perform data preprocessing.

(2) Define the PointNet network shown in Fig. 6, using the cross-entropy loss function as the loss function for classification and segmentation tasks.
(3) Set the optimizer and learning rate, iteratively train the model, and save the trained model after the model converges, that is, a classification model and a segmentation model.
(4) Deploy and test the model, and extract test samples from the constructed test dataset.
(5) Load the previously trained classification model and segmentation model.
(6) Input the test data into the model to obtain the output of the model.
(7) For the classification task, the detection result of whether the blockchain node is suffering from DDoS attack is obtained according to the probability value; For the segmentation task, the type of DDoS attack that the blockchain node is suffering from is obtained based on the probability value.

There are 10,000 samples without DDoS attack categories and 10,000 samples with DDoS attack categories in the created dataset. The total proportion of LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag among the entries with DDoS attack types is 50%, and the number of entries in each sample is uniformly distributed between 4 and 20. To conduct pertinent simulation analysis, we divide the dataset into training, verification, and test sets, with proportions of 70%, 20%, and 10%, respectively. Using an Intel Core i9-13900K @3.00 GHz, 64 GB RAM, and Nvidia GeForce RTX 4090 24G device, the simulations in this section were run. The optimizer selected during the training process is Adam, the initial learning rate is 0.001, and the learning rate attenuation is set to 0.5 every 10 steps, the number of parallel items detected simultaneously is set to 30, the total training algebra is 20, and the batchsize is set to 1024.

First, we simulate the system's classification performance, or its propensity to correctly determine whether a DDoS attack is occurring or not. As stated in the previous section, we constructed an upgraded PointNet network. We also set the maximum number of entries to 30 and the size of the batch training (batch size) to 1024. Fig. 7 displays the loss function curves on the training set and validation set during the training procedure. As can be shown, the proposed classification PointNet network converges after 11 training rounds and obtains a reduced loss function value, which is 0.007744 on the training set and 0.003497 on the validation set, respectively.

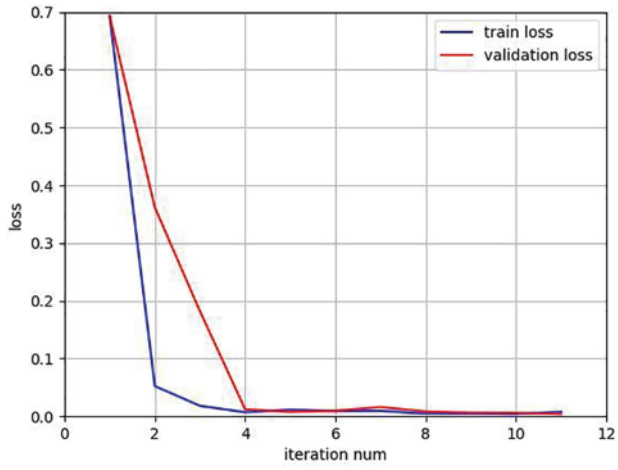

Figure 7: The loss function values of DDoS attack classification

Fig. 8 provides the accuracy rates for the training set, verification set, and test set. Indicating that the proposed DDoS attack detector based on the improved PointNet network has high performance and can Complete the task, it can be seen that with network training, the accuracy rate of the DDoS detection and classification task presents an upward trend, and the final detection accuracy rate It has reached a higher level, namely the training set accuracy rate of 99.71%, the verification set accuracy rate of 99.90%, and the test set accuracy rate of 99.65%.

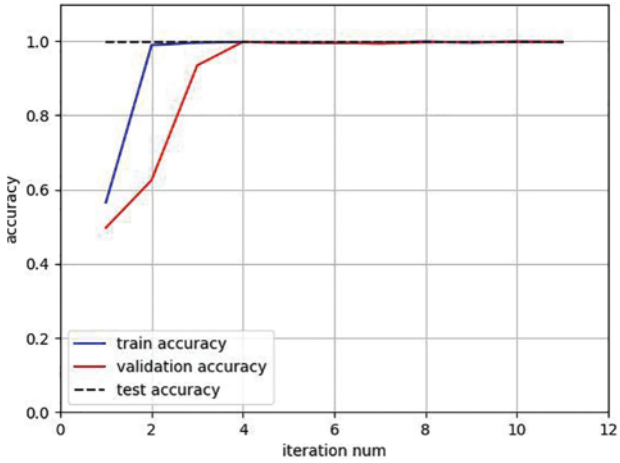

Figure 8: The accuracy results of DDoS attack classification

The improved PointNet segmentation network suggested in the previous part is adopted and trained as we simulate the segmentation task of various forms of DDoS attack detection in the section below. For training on DDoS-affected datasets, we similarly set the batch training size (batchsize) to 1024 and the maximum number of items that can be entered to 30. The convergence curve during the 20 generations training process is shown in Fig. 9. The final loss function values for the training set and verification sets are 0.358323 and 0.366697, respectively, showing that the M-PointNet network has reached convergence. The accuracy rate curve for the DDoS-type segmentation task is shown in Fig. 10 concurrently. The accuracy rates on the training set and verification sets can be observed to gradually increase as the neural network is trained, and the final obtained accuracy rate can be shown to increase as well. It can do superior DDoS-type segmentation tasks based on the accuracy rates of the training set, validation set, and test set, which are 87.38%, 85.44%, and 85.47%, respectively. As shown in Table 2, for the classification performance of DDoS attack detection, the proposed M-PointNet algorithm has a higher classification accuracy, better than LightGBM [38], SVM [39], NB [40]. In addition, in the face of simultaneous and different numbers of DDoS attack feature items, M-PointNet supports the segmentation of each item, and has acceptable results. Table 3 lists the characteristics of these detection methods. It can be seen that M-PointNet is a parallel and accurate detection method when compared with the other listed methods.

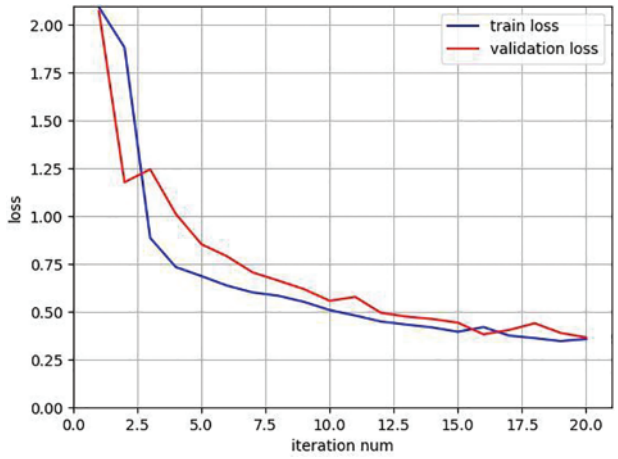

Figure 9: The loss function values of DDoS attack segmentation

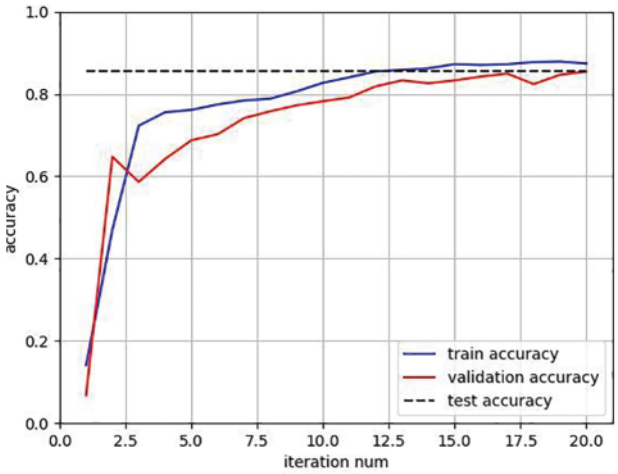

Figure 10: The accuracy results of DDoS attack segmentation

Table 3: An example of the dataset used in this work

Attribute	Classification accuracy	Segmentation accuracy	Characteristics
M-PointNet	99.65%	85.47%	It is a parallel, accurate, and effective detection approach.
LightGBM [38]	99.56%	Not support	It has a low computational cost for higher accuracy.

(Continued)

Table 3: Continued

Attribute	Classification accuracy	Segmentation accuracy	Characteristics
SVM [39]	99.41%	Not support	It is capable of executing tasks such as anomaly-based intrusion detection in real-time.
Naïve Bayes (NB) [40]	95.14%	Not support	It is simple and quick to forecast the test datasets' class.

5 Conclusion

In this study, we design and implement the M-PointNet network for the classification and segmentation of the DDoS attack. Based on the CIC-DDoS2019 dataset, we construct a dataset with several DDoS attack types for network's training, validating, and testing. Our model obtains 99.65% accuracy on the test set when it comes to classifying DDoS attacks, showing that it can accurately separate attack traffic from regular traffic. Our model obtains 85.47% accuracy on the test set in the segmentation task, which indicates its ability to recognize various DDoS attack types. Comparing with methods LightGBM, SVM and Naïve Bayes, we find that our proposed method slightly improves the detection performance of DDoS attacks. In addition, the proposed method supports parallel detection, that is, segmentation of multiple DDoS attack items, which is not available in traditional methods. In conclusion, this study demonstrates that DDoS attacks may be efficiently classified and segmented using the proposed M-PointNet, which provides an approach for network security. Future research will concentrate on enhancing the model's segmentation accuracy and applying it to distributed network environments.

Funding Statement: This work was supported by Hainan Provincial Natural Science Foundation of China (Grant No. 2019RC098, Grant No. 723QN238 and Grant No. 621RC612), National Natural Science Foundation of China (Grant No. 62162022 and 62162024), Key Projects in Hainan Province (Grant ZDYF2020040 and Grant ZDYF2020033), Young Talents' Science and Technology Innovation Project of Hainan Association for Science and Technology (Grant No. QCXM202007).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] T. Wang, C. Zhao, Q. Yang, S. Zhang and S. C. Liew, "Ethna: Analyzing the underlying peer-to-peer network of ethereum blockchain," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2131–2146, 2021.

[2] M. Raikwar and D. Gligoroski, "DoS attacks on blockchain ecosystem," *Euro-Par Workshops*, vol. 13098, pp. 230–242, 2022.

[3] Y. H. Zhang and X. F. Liu, "Traffic redundancy in blockchain systems: The impact of logical and physical network structures," in *2021 IEEE Int. Symp. on Circuits and Systems (ISCAS)*, Daegu, Korea, pp. 1–5, 2021.

[4] H. Zhang, L. Lao, C. Shu and B. Xiao, "Analysis of the communication traffic model for permissioned blockchain based on proof-of-work," in *ICC 2021—IEEE Int. Conf. on Communications*, Montreal, QC, Canada, pp. 1–6, 2021.

[5] R. Shrestha and S. Y. Nam, "Regional blockchain for vehicular networks to prevent 51% attacks," *IEEE Access*, vol. 7, pp. 95033–95045, 2019.

[6] Z. Shah, I. Ullah, H. Li, A. Levula and K. Khurshid, "Blockchain based solutions to mitigate distributed denial of service (DDoS) attacks in the Internet of Things (IoT): A Survey," *Sensors*, vol. 22, no. 3, pp. 1094, 2022.

[7] B. Jia and Y. Liang, "Anti-D chain: A lightweight DDoS attack detection scheme based on heterogeneous ensemble learning in blockchain," *China Communications*, vol. 17, no. 9, pp. 11–24, 2020.

[8] G. Xu, J. Zhang, U. G. Cliff and C. Ma, "An efficient blockchain-based privacy-preserving scheme with attribute and homomorphic encryption," *International Journal of Intelligent Systems*, vol. 37, no. 12, pp. 10715–10750, 2022.

[9] G. Xu, J. Zhang and L. Wang, "An edge computing data privacy-preserving scheme based on blockchain and homomorphic encryption," in *2022 Int. Conf. on Blockchain Technology and Information Security (ICBCTIS)*, Huaihua City, China, pp. 156–159, 2022.

[10] T. V. Le and C. L. Hsu, "A systematic literature review of blockchain technology: Security properties, applications and challenges," *Journal of Internet Technology*, vol. 22, no. 4, pp. 789–801, 2021.

[11] J. Cheng, X. Yao and H. Li, "Cooperative detection method for DDoS attacks based on blockchain," *Computer Systems Science and Engineering*, vol. 43, no. 1, pp. 103–117, 2022.

[12] S. Sahu and A. Verma, "DDoS attack detection in ISP domain using machine learning," in *2019 5th Int. Conf. On Computing, Communication, Control and Automation (ICCUBEA)*, Pune, India, pp. 1–4, 2019.

[13] K. Randhir and R. Tripathi, "Large-scale data storage scheme in blockchain ledger using ipfs and nosql," in *Large-Scale Data Streaming, Processing, and Blockchain Security*, Pennsylvania, US, IGI Global, pp. 91–116, 2021.

[14] P. Kumar, R. Kumar, A. Kumar, A. A. Franklin, S. Garg *et al.,* "Blockchain and deep learning for secure communication in digital twin empowered industrial IoT network," in *IEEE Transactions on Network Science and Engineering*, Piscataway, IEEE, pp. 1–13, 2022.

[15] A. M. S. ElSayed, "Effective deep learning based methods for the anomaly detection in software-defined networks," *University College Dublin. School of Computer Science*, pp. 1–193, 2022.

[16] A. Aljuhani, P. Kumar, R. Kumar, A. Jolfaei and A. K. M. N. Islam, "Fog intelligence for secure smart villages: Architecture, and future challenges," in *IEEE Consumer Electronics Magazine*, Piscataway, IEEE, pp. 1–9, 2022.

[17] R. Fotohi, M. Abdan and S. Ghasemi, "A self-adaptive intrusion detection system for securing UAV-to-UAV communications based on the human immune system in UAV networks," *Journal of Grid Computing*, vol. 20, no. 3, pp. 22, 2022.

[18] P. Kumar, R. Kumar, A. Kumar, A. A. Franklin and A. Jolfaei, "Blockchain and deep learning empowered secure data sharing framework for softwarized UAVs," in *2022 IEEE Int. Conf. on Communications Workshops (ICC Workshops)*, Piscataway, IEEE, pp. 770–775, 2022.

[19] P. Kumar, R. Kumar, G. P. Gupta and R. Tripathi, "A distributed framework for detecting DDoS attacks in smart contract-based Blockchain-IoT Systems by leveraging Fog computing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 6, pp. e4112, 2021.

[20] P. Kumar, R. Kumar, G. P. Gupta, R. Tripathi, A. Jolfaei *et al.,* "A blockchain-orchestrated deep learning approach for secure data transmission in IoT-enabled healthcare system," *Journal of Parallel and Distributed Computing*, vol. 172, pp. 69–83, 2023.

[21] Z. Ashi, L. Aburashed, M. Al-Fawa'reh and M. Qasaimeh, "Fast and reliable DDoS detection using dimensionality reduction and machine learning," in *2020 15th Int. Conf. for Internet Technology and Secured Transactions (ICITST)*, Piscataway, IEEE, pp. 1–10, 2020.

[22] C. Y. Tseung, K. P. Chow and X. Zhang, "Anti-DDoS technique using self-learning bloom filter," in *2017 IEEE Int. Conf. on Intelligence and Security Informatics (ISI)*, Piscataway, IEEE, pp. 204, 2017.

[23] Ö. Kasim, "An efficient and robust deep learning based network anomaly detection against distributed denial of service attacks," *Computer Networks*, vol. 180, no. 4, pp. 107390, 2020.

[24] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," *Journal of Information Security and Applications*, vol. 53, pp. 102532, 2020.

[25] A. Sahi, D. Lai, Y. Li and D. Mohammed, "An efficient DDoS TCP flood attack detection and prevention system in a cloud environment," *IEEE Access*, vol. 5, pp. 6036–6048, 2017.

[26] J. P. A. Maranhão, J. P. C. L. da Costa, E. Javidi, C. B. A. de Andrade and R. T. de Sousa Jr, "Tensor based framework for Distributed Denial of Service attack detection," *Journal of Network and Computer Applications*, vol. 174, no. 6, pp. 102894, 2021.

[27] M. Aamir and S. M. A. Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University-Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, 2021.

[28] A. V. Kachavimath, S. V. Nazare and S. S. Akki, "Distributed denial of service attack detection using naïve bayes and k-nearest neighbor for network forensics," in *2nd Int. Conf. on Innovative Mechanisms for Industry Applications (ICIMIA)*, Piscataway, IEEE, pp. 711–717, 2020.

[29] N. Kumar, A. Aleem and S. Kumar, "Detection of DDoS attack in IoT using machine learning," in *Int. Conf. on Advanced Network Technologies and Intelligent Computing*, Cham, Springer, pp. 190–199, 2021.

[30] I. Eyal, A. E. Gencer, E. G. Sirer and R. van Renesse, "Bitcoin-NG: A scalable blockchain protocol," in *13th USENIX Symp. on Networked Systems Design and Implementation (NSDI 16)*, Santa Clara, CA, pp. 45–59, 2016.

[31] A. Chartuni and J. Márquez, "Multi-Classifier of DDoS attacks in computer networks built on neural networks," *Applied Sciences*, vol. 11, no. 22, pp. 10609, 2021.

[32] O. Osanaiye, H. Cai, K. K. R. Choo, A. Dehghantanha, Z. Xu *et al.,* "Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, pp. 1–10, 2016.

[33] A. Churcher, R. Ullah, J. Ahmad, S. ur Rehman, F. Masood *et al.,* "An experimental analysis of attack classification using machine learning in IoT networks," *Sensors*, vol. 21, no. 2, pp. 446, 2021.

[34] C. R. Qi, L. Yi, H. Su and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5105–5114, 2017.

[35] Z. Liu, H. Tang, Y. Lin and S. Han, "Point-voxel cnn for efficient 3d deep learning," *Advances in Neural Information Processing Systems*, vol. 32, pp. 965–975, 2019.

[36] A. Krizhevsky, I. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[37] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th Int. Symp. on Quality of Service (IWQoS)*, Piscataway, IEEE, pp. 1–2, 2018.

[38] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen *et al.,* "Lightgbm: A highly efficient gradient boosting decision tree," *Advances in Neural Information Processing Systems*, vol. 30, pp. 3149–3157, 2017.

[39] A. Maheshwari, B. Mehraj, M. S. Khan and M. S. Idrisi, "An optimized weighted voting based ensemble model for DDoS attack detection and mitigation in SDN environment," *Microprocessors and Microsystems*, vol. 89, pp. 104412, 2022.

[40] A. Alzahrani and R. J. Alzahrani, "Security analysis of DDoS attacks using machine learning algorithms in networks traffic," *Electronics*, vol. 10, no. 23, pp. 2919, 2021.