Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.045588

ARTICLE

An Adaptive DDoS Detection and Classification Method in Blockchain Using an Integrated Multi-Models

Xiulai Li[1,2,3,4], Jieren Cheng[1,3,*], Chengchun Ruan[1,3], Bin Zhang[1,3], Xiangyan Tang[1,3] and Mengzhe Sun[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]School of Information Science and Technology, Qiongtai Normal University, Haikou, 570228, China

*Corresponding Author: Jieren Cheng. Email: cjr22@163.com

Received: 31 August 2023 Accepted: 03 November 2023 Published: 26 December 2023

ABSTRACT

With the rising adoption of blockchain technology due to its decentralized, secure, and transparent features, ensuring its resilience against network threats, especially Distributed Denial of Service (DDoS) attacks, is crucial. This research addresses the vulnerability of blockchain systems to DDoS assaults, which undermine their core decentralized characteristics, posing threats to their security and reliability. We have devised a novel adaptive integration technique for the detection and identification of varied DDoS attacks. To ensure the robustness and validity of our approach, a dataset amalgamating multiple DDoS attacks was derived from the CIC-DDoS2019 dataset. Using this, our methodology was applied to detect DDoS threats and further classify them into seven unique attack subcategories. To cope with the broad spectrum of DDoS attack variations, a holistic framework has been proposed that seamlessly integrates five machine learning models: Gate Recurrent Unit (GRU), Convolutional Neural Networks (CNN), Long-Short Term Memory (LSTM), Deep Neural Networks (DNN), and Support Vector Machine (SVM). The innovative aspect of our framework is the introduction of a dynamic weight adjustment mechanism, enhancing the system's adaptability. Experimental results substantiate the superiority of our ensemble method in comparison to singular models across various evaluation metrics. The framework displayed remarkable accuracy, with rates reaching 99.71% for detection and 87.62% for classification tasks. By developing a comprehensive and adaptive methodology, this study paves the way for strengthening the defense mechanisms of blockchain systems against DDoS attacks. The ensemble approach, combined with the dynamic weight adjustment, offers promise in ensuring blockchain's enduring security and trustworthiness.

KEYWORDS

Blockchain; DDoS; multi-models; adaptive detection

1 Introduction

Blockchain, an innovative distributed and decentralized digital ledger technology, has revolutionized numerous sectors due to its ability to systematically record transactions and ensure data consistency without relying on a centralized entity [1]. This technology stores data in blocks connected in a linear, chronological sequence [2]. One of its strengths lies in its decentralized architecture, where each node possesses an identical copy of the ledger, eliminating central points of failure and providing robustness [3]. Cryptographic hashing bolstered its inherent transparency and security features, making data manipulation nearly impossible once recorded [4]. Although every transaction is visible to all network nodes, blockchain offers transactional privacy by allowing participants to employ anonymous addresses. Moreover, the rise of blockchain has seen the introduction of smart contracts automated contractual mechanisms triggered by specific conditions, broadening its application domain [5]. Data consistency across nodes is achieved using a consensus mechanism, a pivotal component that validates adding new blocks to the chain [6].

Data consistency across nodes is achieved using a consensus mechanism, a pivotal component that validates adding new blocks to the chain [7]. While advantageous, the decentralized nature of blockchain technology also attracts cyber attackers aiming to exploit its structure and compromise its security using DDoS assaults. In blockchain's vast and dynamic data streams, capturing intricate DDoS attack patterns using a single detection model is inherently challenging. Undetected or inadequately addressed attacks can have disastrous consequences, both financially and reputationally. The industry's key challenges revolve around developing detection techniques that are precise, adaptable, and able to seamlessly combine various models for a comprehensive defense against DDoS attacks [8]. Attackers initiate attacks by manipulating a substantial quantity of compromised computers, frequently in a decentralized fashion, to amplify the attack's magnitude [9]. These assaults can potentially result in network congestion, increased latency, and service disruptions and present a significant risk to the availability and performance of the network [10]. Hence, the study and safeguarding of DDoS assaults has significant significance.

Blockchain technology has been widely adopted in many industries due to its decentralized nature, security, and transparency. However, its popularity has also become a target for cyber attackers, especially DDoS attacks. These attacks aim to weaken the decentralized structure of blockchain, reducing its security and stability. Blockchain data is complex and always changing. This makes it hard for a single detection system to identify and respond to sophisticated attack patterns. If these DDoS attacks are not detected and stopped, they can completely shut down a system. This leads to financial losses and damages the reputation of users and organizations involved.

There are several challenges in this field:

1. DDoS attack detection must be achieved more quickly and accurately, even as the data changes.

2. The detection models are not supposed to become too specialized so they can handle new types of attacks.

3. Different models must combine and respond effectively to DDoS attacks, using each model's strengths best.

2 Related Work

Machine learning has emerged as a formidable defense against DDoS attacks. It is proficient in extracting critical network traffic characteristics, swiftly adapting to novel attack patterns, and delivering high-precision real-time responses. Such capabilities reduce the need for human intervention and ensure continuous learning and optimization, making machine learning a pivotal tool in the cybersecurity landscape [11].

In blockchain ecosystems, machine learning, being inherently data-centric, showcases considerable promise. With their complex and ever-evolving nature, Blockchain networks often stump traditional detection techniques. However, with machine learning's adeptness at using historical data to differentiate between regular and anomalous traffic patterns, the timely identification of DDoS assaults becomes feasible [12,13]. This adaptability ensures that the system remains consistent in its performance, even when faced with the evolving tactics of attackers.ng techniques employed by attackers, hence ensuring consistent and effective detection performance.

The academic realm, too, acknowledges the potency of machine learning in detecting DDoS assaults, especially within the blockchain technology framework [14]. There is an anticipated rise in machine learning techniques, given their potential to heighten detection accuracy and ensure real-time mitigation of the negative impacts of assaults on blockchain networks [15]. The inherent automation in machine learning also underscores its potential to diminish human involvement and curtail related expenses [16].

Deep learning, a subset of machine learning, amplifies these capabilities. As seen in [17], deep learning can extract and process features autonomously, ensuring robust representation. While DDoS attacks aim to incapacitate applications [18], strategies like the one proposed in [19], which blends blockchain technology with deep learning, can significantly elevate the defense mechanisms of public transport systems against cyber threats. The efficacy of such approaches is evident, with F1 scores consistently surpassing the 95% mark.

Innovative approaches like integrating virtual reality anti-DDoS chains, distributed detection frameworks, and hybrid learning have been proposed to tackle new-age DDoS threats in blockchain networks [10]. Techniques like AdaBoost and Random Forest have showcased their prowess in discerning DDoS attack patterns in P2P networks. The suggested approach in [20] addresses the limitations of single-target detection and privacy concerns by utilizing federated learning multi-domain DDoS detection. The implementation of blockchain-based reputation assessment is proposed to enhance the resilience and reliability of existing systems. The experimental results demonstrate a precision level above 95%. Reference [21] presented a proposal for integrated solutions that leverage Software-Defined Networking (SDN), blockchain technology, and machine learning techniques. The aim is to enhance the security of Internet of Things (IoT) systems and bolster the detection and mitigation of DDoS attacks. The LSTM-CGAN method produced LDDoS adversarial samples to improve the detection model's robustness for blockchain wireless networks [22]. In [23], integrating blockchain technology and machine learning algorithms is suggested to establish a layered architecture for safe applications in the Industrial Internet of Things (IIoT) within the context of intelligent manufacturing. This architecture aims to facilitate attack detection and sensor access control. In [24], a blockchain-based permission system that utilizes lightweight technology uses the quorum Physical Unclonable Function (PUF) model to secure key pairs of IoT devices. The collaborative detection system using machine learning integration technology detects DDoS attacks on IoT devices, providing lower false positive rates and better detection rates. At the same time, through the blockchain system, alarm information is securely shared with all nodes of the IoT network. Reference [25] presented a novel approach utilizing deep learning techniques with optimization methods, blockchain technology, and smart contracts to enhance the detection and mitigation of DDoS assaults. This study proposes the integration of user requests, smart contract verification, and using Poaching Raptor to optimize DNN designed explicitly for attack detection purposes. The approach demonstrated a performance of 96.3% in recall, 98.22% in precision, 3.33% in false alarm rate, and 95.12% in accuracy rate. Reference [26] introduced a novel approach to asynchronous federated learning using blockchain technology. The suggested system incorporates a blockchain-based asynchronous federated learning scheme with a dynamic scaling factor (DBAFL), to enhance the efficiency, reliability, and performance of continuously updating traffic prediction models in the context of smart public transportation. Reference [27] introduced a novel intrusion detection system (IDS) inside the framework of distributed fog computing. The primary objective of this IDS is to identify and mitigate DDoS assaults, specifically targeting mining pools in blockchain-based IoT networks. The verification of model validity involves the training of Random Forest and eXtreme Gradient Boosting (XGBoost) algorithms on distributed fog nodes. Based on the analysis conducted using the BoT-IoT dataset, it is evident that XGBoost outperforms in the context of two-class attack detection. In contrast, Random Forest exhibits superior performance in multi-classification attack detection. In response to the issue of DDoS attacks in the context of the IIoT, reference [28] presented a novel approach for mitigating such attacks. The suggested approach involves the integration of a multi-point collaborative defense mechanism, which comprises an edge detection mechanism and a blockchain-based collaborative defense model. The implementation of blockchain technology facilitates defensive information sharing. Additionally, the proposed rapid-sharing method effectively mitigates the latency in information propagation. The simulation results confirm the efficacy of the edge detection and quick-sharing mechanism.

Diving into ensemble learning, it serves as a beacon in machine learning that aims to bolster performance by amalgamating predictions from various foundational models. The essence of ensemble learning is to counteract overfitting and magnify a model's robustness and accuracy. Studies such as [29–31] have delved into this domain, presenting innovative ensemble learning methods that have showcased remarkable efficiency and accuracy over traditional models.

Machine learning has significantly enhanced DDoS attack detection in blockchain systems. However, its full potential still needs to be explored, primarily due to the complexity of blockchain network traffic and insufficient integration of ensemble learning methods in research. Given the evolving nature of these attacks, traditional detection methods often need to catch up. Despite isolated studies highlighting machine learning's capability in this realm, a comprehensive ensemble-based strategy remains elusive. Our work addresses this gap by introducing a robust ensemble learning method, combining GRU, CNN, LSTM, DNN, and SVM models. This integrated approach promises superior accuracy in detecting DDoS patterns over individual models, particularly in the intricate environment of blockchain networks. With the added advantage of dynamic weight adjustments, our system can swiftly adapt to changing network scenarios, ensuring consistent defense against ever-evolving DDoS tactics. In essence, by marrying ensemble learning with dynamic adjustments, we aim to elevate the resilience and security of blockchain systems against DDoS threats.

3 Background

3.1 Blockchain

Blockchain technology works as a distributed and decentralized digital ledger, which has been later utilized in the context of cryptocurrencies like Bitcoin [32]. The fundamental premise involves establishing a connection between transactional data, represented as blocks, to construct a continuously expanding chain-like structure. Encryption in blockchain technology guarantees the integrity and confidentiality of data. At the same time, the absence of a central authority mitigates the vulnerability associated with a singular point of failure commonly found in conventional centralized systems [33].

In recent years, the technology known as blockchain has garnered significant attention and practical implementation due to its disruptive nature. This has resulted in substantial academic interest in the topic. The fundamental concept underlying this initiative is the establishment of a decentralized, safe, and reliable platform for storing and selling data, incorporating several distinctive attributes, which can be summarized as follows:

(1) The decentralized nature of blockchain technology eliminates the need for intermediaries in traditional centralized systems, allowing for direct peer-to-peer connection. This enhances transaction efficiency and mitigates the risks and costs associated with intermediary connections, enhancing the efficiency and transparency of data sharing.

(2) The blockchain's transparency and immutability ensure data security and dependability. Every transaction and piece of information is systematically organized into a block in the blockchain system. Each block is designed to include the data from the preceding block, resulting in an interconnected chain structure that cannot be separated. Encryption and consensus procedures enable individuals to authenticate and trace the chronological sequence of a specific transaction, hence ensuring the reliability and credibility of transactional information.

(3) The smart contract features of the blockchain enable automated execution. Smart contracts refer to pre-programmed code enabling operations automation by specific criteria, hence obviating third-party intervention. This presents novel opportunities for various business and transaction circumstances, potentially decreasing transaction expenses and expediting the transactional procedure.

3.2 Dataset and DDoS Attack Categories Selected

The CIC-DDoS 2019 dataset holds substantial importance within the realm of network security. As a research tool, it significantly advances the detection of DDoS attacks. The distinctiveness of this system resides in the intricate interplay between legitimate and malicious network traffic, which offers valuable data for the training and validation of deep learning models. The dataset holds significant importance as it enables researchers and practitioners to investigate novel detection techniques for effectively discerning tiny deviations between regular network traffic and potential DDoS attack traffic. By conducting a comprehensive analysis of this data from several dimensions, it is anticipated that we would acquire a profound understanding of cyber defense and threat response tactics, thus enhancing the network's capacity to counter attacks and elevating its overall level of security.

This work employed the CIC-DDoS2019 dataset to conduct research utilizing machine learning techniques to analyze a range of standard and pathological behaviors, including BENIGN, Lightweight Directory Access Protocol (LDAP), MSSQL, NetWork Basic Input/Output System (NetBIOS), Portmap, Syn, UserDatagramProtocol (UDP), and UDPLag [34]. We describe various types of DDoS attacks.

(1) LDAP: The LDAP protocol may be utilized by hostile actors to engage in nefarious operations, such as the perpetration of LDAP injection attacks. The present attack exploits the utilization of LDAP query structures to carry out malevolent activities that can result in many concerns, including but not limited to the revealing of sensitive information, bypassing authentication mechanisms, or causing a denial of service.

(2) MSSQL: This attack capitalizes on a weakness inside an application, wherein it inserts malicious SQL queries into database activities. This has the potential to result in unauthorized access and release.

(3) NetBIOS: NetBIOS has the potential to be implicated in perpetuating NetBIOS Name Service (NBNS) spoofing attacks. Adversaries falsify genuine hosts by transmitting counterfeit NBNS answers, potentially facilitating man-in-the-middle attacks or pilfering sensitive information.

(4) Portmap: Portmap has been identified as potentially associated with Remote Procedure Call (RPC) attacks. An assailant can use the RPC vulnerability to transmit malevolent requests, perhaps resulting in unauthorized entry or the execution of remote code.

(5) Syn: The attack leverages the vulnerability present in the Transmission Control Protocol (TCP) protocol to deplete server resources and induce service unavailability. This is achieved by inundating the server with a substantial volume of incomplete connection requests during the handshake process.

(6) UDP: Transmitting a significant quantity of UDP packets enables the assailant to forgo establishing a connection. This results in an excessive burden on the server, thereby impacting the provision of services.

(7) UDPLag: In contrast to conventional UDP attacks, the UDPLag attackers employ the source IP address spoofing technique to induce response traffic directed towards the target of the attack, hence intensifying the impact of the attack. Various strategies can be employed to counteract UDPLag attacks.

4 An Adaptive DDoS Attack Detection Based on Multiple Models

In this section, Section 4.1, a comprehensive preprocessing of the CIC-DDoS2019 dataset is conducted. The dataset is carefully examined, specific data instances are chosen and organized, and appropriate transformations and filters are applied. Following data preparation, Section 4.2 comprehensively describes five distinct single models specifically tailored to extract features and recognize patterns associated with DDoS attacks. These models include GRU, CNN, LSTM, DNN, and SVM. Nevertheless, owing to the constraints associated with employing a solitary model to depict diverse DDoS attacks, the third section of this paper introduces a novel approach for detecting such attacks. This approach integrates five distinct models on each blockchain node, enhancing the detection process's precision and dependability. Furthermore, to enhance the precision of classification, Section 4.4 presents an adaptive multi-model classification approach that employs a soft voting mechanism to average the prediction probabilities of each model. This technique aims to enhance the overall prediction outcome. Fig. 1 is the schematic diagram.

4.1 Data Pre-Process

This study's data selection process adheres to the criteria above, wherein 10,500 instances labeled as BENIGN are chosen. Additionally, 1,500 pieces from each of the following categories, LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag, are utilized as inputs for the model model. As depicted in Fig. 2, the left pie chart equitably represents data of instances that were attacked *vs*. those that were not, each accounting for 50%. The right chart provides a balanced delineation of seven distinct attack types—LDAP, MSSQL, NetBIOS, Portmap, Syn, UDP, and UDPLag—each constituting 14.3% of the total dataset. This comprehensive distribution ensures the model's unbiased generalization across diverse cyber threats. In the process of file loading, we have devised two approaches: large file loading and small file loading. Specifically, when the model input interface identifies a file of considerable size, it will be loaded in segments based on the timestamp. This approach offers the benefits of enhancing model efficiency and redundancy. Several labels, such as "Infinity", "np.inf", and "nan", have been substituted by zeros. Moreover, it is necessary to convert the formats of "Flow Packets/s" and "Flow Bytes/s". The correlation was computed, followed by the removal of invalid columns. The decision tree filtering method was then employed to eliminate 14 features from the initial 87 feature quantities in the CIC-DDoS2019 dataset [35]. The features are categorized based on their level of significance. The overall count encompasses the subsequent attributes: The features of interest in this context include the 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, 14 features in total. Ultimately, we successfully conserved and modernized the timestamps to facilitate subsequent modeling utilizing time-series processing techniques.

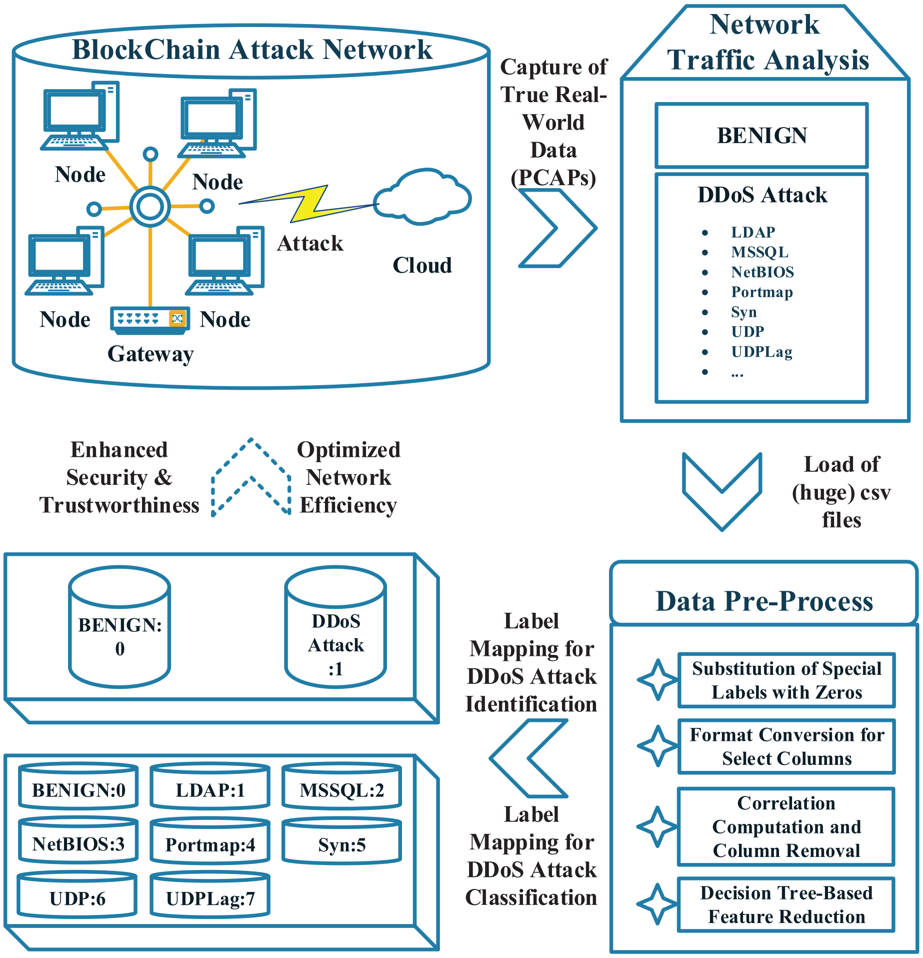
BlockChain Attack Network
Node
Node
Node
Node
Gateway
Attack
Cloud
Capture of True Real-World Data (PCAPs)
Network Traffic Analysis
BENIGN
DDoS Attack
LDAP
MSSQL
NetBIOS
Portmap
Syn
UDP
UDPLag
...
Load of (huge) csv files
Data Pre-Process
Substitution of Special Labels with Zeros
Format Conversion for Select Columns
Correlation Computation and Column Removal
Decision Tree-Based Feature Reduction
Label Mapping for DDoS Attack Identification
BENIGN: 0
DDoS Attack :1
Label Mapping for DDoS Attack Classification
BENIGN:0
LDAP:1
MSSQL:2
NetBIOS:3
Portmap:4
Syn:5
UDP:6
UDPLag:7
Enhanced Security & Trustworthiness
Optimized Network Efficiency

Figure 1: Schematic diagram

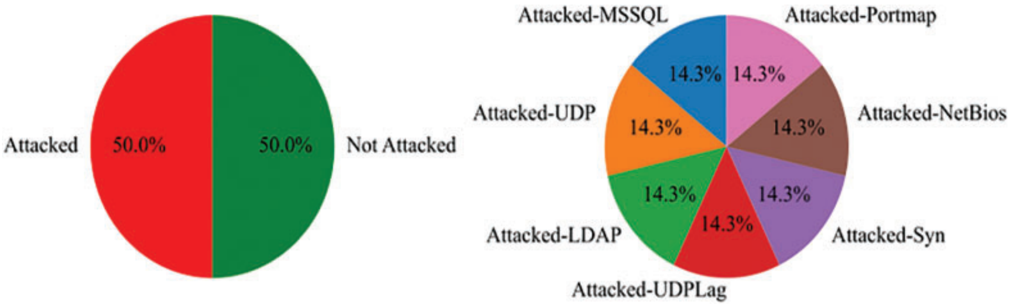

Figure 2: Data division scale chart for detection and classification

Correlation computations were employed to assess the association between data features. In light of the provided information, specific columns deemed invalid or unnecessary were removed to enhance the model's efficacy and precision while reducing its complexity.

4.2 Single-Model Detection and Classification Methods

In the rapidly evolving domain of cyber-attacks, the detection and mitigation of DDoS attacks necessitate models that can handle vast amounts of data, recognize subtle patterns, and adapt over time. Traditional methods, while foundational, are not fully equipped to handle the dynamic nature of DDoS attacks and their rapidly changing patterns. The models chosen for this study—GRU, CNN, LSTM, DNN, and SVM—represent a blend of classical machine learning and modern deep learning techniques. Each of these models offers distinct advantages tailored for DDoS detection.

GRU and LSTM, as variants of recurrent neural networks (RNNs), excel in processing sequence data, capturing long-range dependencies, and retaining memory of past events, which is crucial for analyzing continuous network traffic. CNN excels at automatically extracting hierarchical features from raw data, especially useful when the nature of the attack can be discerned from local patterns. DNN provides a deep, layered approach to model complex non-linear relationships in the data, ensuring that subtle attack patterns hidden in vast traffic data are captured. SVM, a classical machine learning algorithm, shines in its ability to distinctly separate attack traffic from normal traffic, ensuring a clear demarcation for detection.

Combining the strengths of these models allows us to create a comprehensive DDoS detection system that not only addresses the limitations of traditional methods but also leverages the benefits of state-of-the-art techniques. This holistic approach promises improved detection rates and reduced false positives, vital for maintaining the integrity and efficiency of network systems.

4.2.1 GRU

The GRU is a modified version of the RNN used for sequence modeling. It addresses the gradient vanishing issue in regular RNNs when dealing with extended sequences and offers improved modeling capabilities. The GRU incorporates a gating system that facilitates the management of memory cell states by dynamically controlling the gating unit. This mechanism enables the GRU to efficiently capture long-range dependencies within a sequence by facilitating the renewal and forgetting of memory cell states [36].

The primary idea behind the GRU is the use of two locking units, called the Reset Gate and the Update Gate. These two gating methods control how much the previous hidden state and the current input are fused and check if the previous hidden state needs to be changed to match the current hidden state.

The calculation for the reset door (*r_t*) is as follows:

$$r_t = \sigma\left(W_{ir} \cdot x_t + b_{ir} + W_{hr} \cdot h_{t-1} + b_{hr}\right) \tag{1}$$

Among the variables under consideration, x_t represents the input at present, h_{t-1} denotes the hidden state at the preceding time step, W_{ir} and W_{hr} correspond to the weights associated with the input and hidden states while b_{ir} and b_{hr} represent the biases.

The formula for calculating the update gate (*z_t*) is as follows:

$$z_t = \sigma\left(W_{iz} \cdot x_t + b_{iz} + W_{hz} \cdot h_{t-1} + b_{hz}\right) \tag{2}$$

The calculation of the candidate concealed state $\tilde{h}_t$ can be achieved by the process of resetting and updating the gate.

$$\tilde{h}_t = \tanh\left(W_{ic} \cdot x_t + b_{ic} + r_t \cdot (W_{hc} \cdot h_{t-1} + b_{hc})\right) \tag{3}$$

The calculation of the hidden state h_t at the current time step is performed by updating the gate.

$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t \tag{4}$$

This study presents the development of a GRU network model specifically designed to detect and mitigate DDoS attacks.

The GRU model in this paper consists of an input layer, which maps the input sequence as a low-dimensional vector, and a GRU layer, which has a unit of 64 hidden neurons to efficiently capture the critical features of the sequence. This is followed by a dense layer that maps the result to [0,1] via the Sigmoid function to perform detection. In the training stage, based on the binary_crossentropy loss function and Adam optimizer, the model is iteratively trained for ten epochs on the embedded X_train sequence and the corresponding y_train labels, and each batch contains 64 samples so that the model can sequence data analysis and detection tasks.

4.2.2 CNN

CNNs work by automatically pulling out features from raw data using convolutional procedures. For CNNs, convolution processes are the most essential building blocks. The input data is put through the convolution process to get local features [37]. This is achieved by sliding a convolution kernel across the input data. The process of obtaining the output feature map involves the element-wise multiplication of the convolution kernel with each local area of the input data, followed by the summation of the results. This procedure can capture the local properties of DDoS data.

The formula used to calculate convolution operations is as follows:

$$S(i,j) = (I * K)(i,j) = \sum_1^m \sum_1^n I(i+m, j+n) \cdot K(m,n) \tag{5}$$

$S(i,j)$ represents the pixel value of the output feature map, $I(i+m,j+n)$ represents the pixel value of the input data, and $K(m,n)$ represents the weight of the convolution kernel.

We design a CNN net model with the development of the structure of multi-layer convolutional and pooling layers. The CNN model designed in this paper mainly comprises a convolutional layer, pooling layer, flattening layer and dense layer, which is used for detection tasks. First, the model's input is an image with the number of features, and the sequence length, in which the convolutional layer uses 32 convolution kernels of size three and the ReLU activation function to extract the image features. Next, a pooled kernel of size two is used for maximum pooling to reduce the size of the feature map. This is followed by 64 convolution kernels of size three and the ReLU activation function for convolution operations, and maximum pooling is performed again. After feature extraction, the multidimensional feature map is converted into a one-dimensional vector by flattening the layer to prepare for subsequent classification. The final dense layer contains a neuron and applies the Sigmoid activation function, mapping the output to the range of [0,1] for detection with these steps. The training phase of the model uses the binary_crossentropy loss function and the Adam optimizer. Ten epochs were trained on the training set X_train_cnn and corresponding label y_train, with a size 64 per batch. The model can learn to extract features from the input image and perform efficient DDoS detection. The pooling layer decreases the data's dimensionality by downsampling the input feature map. This downsampling technique allows for the retention of crucial information while also minimizing the computational workload.

4.2.3 LSTM

The LSTM model is good at spotting patterns in network traffic data, helping to find potential attacks. It uses a unique gate in its structure, allowing it to remember critical long-term patterns and ignore unnecessary input. This approach boosts the accuracy of detecting attacks. The calculation formula for Forget Gate is as follows:

$$f_t = \sigma \left(W_f \cdot [h_{t-1}, x_t] + b_f \right) \tag{6}$$

In the given formula, the hidden state of the previous moment is denoted as h_{t-1}, the input at the present moment is denoted as x_t, the weight is denoted as W_f, the bias is denoted as b_f, and the sigmoid function is denoted as σ.

The calculation formula for Input Gate is as follows:

$$i_t = \sigma \left(W_i \cdot [h_{t-1}, x_t] + b_i \right) \tag{7}$$

$$\tilde{C}_t = \tanh \left(W_C \cdot [h_{t-1}, x_t] + b_C \right) \tag{8}$$

where i_t is the Input Gate, $\tilde{C}_t$ is a new candidate cell state.

By utilizing the forgetting gate and the input gate, it becomes possible to compute a fresh cell state C_t.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{9}$$

The calculation formula for Output Gate is as follows:

$$o_t = \sigma \left(W_o \cdot [h_{t-1}, x_t] + b_o \right) \tag{10}$$

Ultimately, calculating the hidden state at the moment is achieved by generating of the gate and cell state outputs h_t.

$$h_t = o_t \cdot \tanh \left(C_t \right) \tag{11}$$

In DDoS detection, the LSTM model analyzes network traffic data as an input sequence. This model effectively learns and identifies patterns within the traffic data by iteratively processing the inputs at each sequential time step [38]. The LSTM network changes the state of its cells dynamically in reaction to changes in the flow data that comes in. This is done to look for possible signs of attacks. By teaching the LSTM model, the neural network can learn about standard traffic patterns and quickly spot traffic cases that do not follow these patterns. Such changes could mean strange traffic patterns, a sign of DDoS attacks. Its ability to handle temporal data sequences well and successfully record long-term interdependencies makes LSTM useful for finding DDoS attacks. With LSTM models, you can use gating methods to control the data flow. These models benefit cybersecurity because they can find possible security holes independently.

The development of the LSTM model, as outlined in this study, is structured as follows: the input layer resembles to the GRU model since it receives input data and subsequently transforms it into low-dimensional vectors. Subsequently, the LSTM layer is employed, comprising a unit of 64 hidden neurons that effectively capture long-term dependencies within sequential data. Following the preceding layer, a densely connected layer consists of a single neuron. This neuron employs the Sigmoid activation function to transform the output values into the interval [0,1]. This specific activation function is commonly utilized in DDoS detection tasks. Regarding the training process, ten epochs were executed on the training dataset, denoted as X_train, along with its associated labels, y_train. The binary_crossentropy loss function and the Adam optimizer were employed during this training procedure. Each batch consisted of 64 samples. Through this architectural design, the model undergoes training to comprehend and recognize extended patterns within sequential data, enabling it to execute DDoS detection tasks effectively.

4.2.4 DNN

The fundamental concept underlying DNN is the automated extraction of high-level features using multi-layer neural networks. This enables the nonlinear modeling of network traffic data, facilitating the identification of potential attack patterns [39]. DNNs are composed of several hidden layers, with each layer including numerous neurons. In a DNN, every individual neuron takes input from the preceding layer of neurons and executes a nonlinear mapping using the activation function. The resulting output is then transmitted to the subsequent layer of neurons. DNNs can acquire higher-level feature representations from the initial input data using a complex, multi-layered cascading structure.

The mapping of input layers to one hidden layer can be expressed as:

$$z = W_1 \cdot x + b_1 \tag{12}$$

$$h = \sigma\left(z\right) \tag{13}$$

where x represents the input data, W_1 and b_1 represent the weight and bias, respectively, and σ represents the activation function.

The mapping from hidden layer to output layer is:

$$y = W_2 \cdot h + b_2 \tag{14}$$

In DNNs, the weights and biases are changed repeatedly with the backpropagation method. The loss function is made as small as possible to make this change, which makes the model work better overall. To see if there is a DDoS attack, you need to put in data that shows how network traffic works. The process's goal is to produce a result that states whether the observed network data is an indication of an attack or not. Because a deep neural network can map data in a way that is not linear, it is possible to find complicated patterns and features that can be used to spot DDoS attacks. DNNs can make it easier for the model to explain itself and get better feature extraction by adding more hidden layers. With a lot of training data, DNNs can learn on their own about different kinds of threats. In short, DNN uses neural networks' multi-layer structure to automatically pull out complex traits from network traffic data. This makes it possible to find and sort DDoS strikes into different groups. As a spy tool, this one is useful because it can handle nonlinear modeling well.

The DNN model presented in this study incorporates an input layer that consists of a neuron count equivalent to the number of features, hence facilitating the reception of input data. The subsequent architecture consists of two concealed layers, each comprising 64 and 32 neurons, respectively. These layers are responsible for incorporating non-linear characteristics into the model by utilizing the rectified linear unit (ReLU) activation function, hence augmenting the model's ability to represent complex relationships. The output layer is comprised of a single neuron that employs the Sigmoid activation function in order to transfer the output values to a range between 0 and 1. Regarding the training process, the model employs the "Adam" loss function, undergoes a maximum of 100 iterations, and utilizes the training dataset denoted as X_train and y_train. By utilizing these specified configurations, the model demonstrates the capability to acquire knowledge about patterns within the dataset, hence enabling it to achieve precise DDoS detection.

4.2.5 SVM

The principle of SVM is to separate different categories of data by building a hyperplane to achieve the classification of attack traffic and normal traffic. The fundamental concept behind SVM is the identification of a hyperplane that maximizes the distance between the hyperplane and the nearest data point, also known as the support vector. In the context of a DDoS detection task, the objective of SVM is to identify an ideal hyperplane that may be mathematically represented as [40]:

$$w^T x + b = 0 \tag{15}$$

where w is the normal vector and b is the bias, w and b are optimized by classifying the data points. The mathematical principle of the SVM can be represented by the following formula, where yi denotes the label assigned to the data point (with 1 indicating an attacked instance and 0 indicating an instance that is not attacked).

$$\begin{cases} y_i \left(w^T x_i + b \right) \geq 1 \quad (attack) \\ y_i \left(w^T x_i + b \right) \leq 0 \;\; (no - attack) \end{cases} \tag{16}$$

To find DDoS attacks, SVMs learn about a hyperplane and can tell the difference between attack data and normal traffic. The SVM algorithm is used to turn data about network traffic into a feature space with many variables. Finding a hyperplane in the feature space that successfully separates and distinguishes between different types of data is part of this process. By building a hyperplane, the SVM algorithm can tell the difference between DDoS attack traffic and normal traffic. The widening of the interval makes it possible to classify things well even when the data is spread out in a complicated way. This makes it a powerful tool for finding DDoS attacks.

The optimization goal of SVM is to maximize the interval, that is, to maximize $\frac{2}{\|w\|}$, equivalent to minimize $\|w\|^2$, w and b can be solved by solving the Lagrange multiplier method to obtain the duality problem.

The SVM model employed in this study utilizes linear kernels and is trained using training datasets (X_train and y_train) using linear kernels and the parameter “probability = True”. Regarding performance evaluation, the test set’s true label and the predicted label generated by the SVM model were thoroughly assessed.

4.3 Integrated Multi-Model Detection Method

Ensemble learning forms the bedrock of our research’s theoretical framework, underscoring the idea that the amalgamation of multiple models can yield predictions superior in accuracy and reliability than a singular model. This concept, which synergistically integrates diverse learning algorithms or multiple iterations of a similar algorithm, offers a fortified prediction model, making it particularly vital given the multifaceted and evolving nature of DDoS attacks. A singular model approach can inadvertently overlook certain nuances, creating vulnerabilities; however, ensemble learning, rooted in diversifying model perspectives, ensures comprehensive coverage against a vast array of cyber threats. Additionally, the ensemble framework inherently operates on consensus-building, a principle where individual model outputs culminate in a weighted, collective decision. This resonates seamlessly with the decentralized consensus mechanisms inherent in blockchain, further solidifying ensemble learning as an optimal choice for our proposed DDoS detection within blockchain networks. In harnessing the collective prowess of multiple models through this theoretical lens, we bolster detection accuracy and craft a system primed for resilience against the dynamic challenges posed by DDoS threats.

This research paper presents a new approach for detecting DDoS attacks. The proposed method involves integrating the detection mechanism into blockchain networks, aiming to enhance the precision and dependability of the detection process. The limitations of traditional single models in effectively representing the various forms of DDoS attacks have prompted us to develop a set of five distinct specialized models on each blockchain node. In the event of a network assault, individual nodes operate autonomously, executing their own models and afterward employing a hard voting process to ascertain the presence of a DDoS attack. The use of a multi-model ensemble technique has exhibited considerable advancement in augmenting the accuracy of detection. The reliability and resilience of detection can be enhanced by capturing attack fingerprints from several dimensions through the joint efforts of various models [38]. The assumption is made that the integration of models can enhance performance in a linear manner without considering potential redundancies or complementarities that may exist between the models.

In contrast to conventional centralized detection methods, our proposed methodology incorporates an ensemble model within a blockchain network. One notable benefit of this approach is its ability to fully leverage the decentralized nature of blockchain technology. Every node operates autonomously, mitigating the potential vulnerability of a singular point of failure. In the event of an attack, nodes achieve consensus through a process of hard voting, hence bolstering the confidence of the detection outcomes.

Furthermore, the novelty of this approach is also evident in the development of customized models for DDoS attacks. Every model has been adjusted to cater to its specific requirements, hence enhancing its adaptability of DDoS attack detection. The use of this particular design enables our approach to showcase its superiority in terms of both detection accuracy and efficiency.

As shown in Fig. 3, this paper uses the five learning models designed above to detect DDoS attacks, which can be summarized as follows:

$$\hat{y}_{\text{ensemble}} = argmax_j \sum_{i=1}^{N} \mathbb{I}\left(\hat{y}_i = j\right) \tag{17}$$

In this paper, $N = 5, \hat{y}_{\text{ensemble}}$ represents the final prediction result of the ensemble model for a sample DDoS attack. j has a selection of 1 of 0, which means the model detects the attack or not. $argmax_j$ means the result referred to the max count of the responding prediction of each model. $\hat{y}_i$ represents the prediction result of the ith machine learning model. j is the label of the possible result $\mathbb{I}\left(\hat{y}_i = j\right)$ is an indication function, when $\hat{y}_i = j$, it is 1, otherwise is 0.

4.4 Adaptive Multi-Model Classification Method

The ensemble classification model for DDoS assaults incorporates multi-model techniques in order to enhance the resilience and precision of the model. In the initial phase, a hard voting method was employed, which disregards the model's projected probability and instead makes judgments solely based on the label category of the final prediction. The categorization label for each sample was established by applying the principle of "majority vote" to the labels provided by each underlying model. Nevertheless, this particular methodology fails to consider the amount of confidence associated with the predictions made by the model. In order to address this issue, we propose the implementation of a soft voting mechanism. This mechanism enhances the accuracy of the final prediction outcome by averaging the probability distribution of predictions from each individual model.

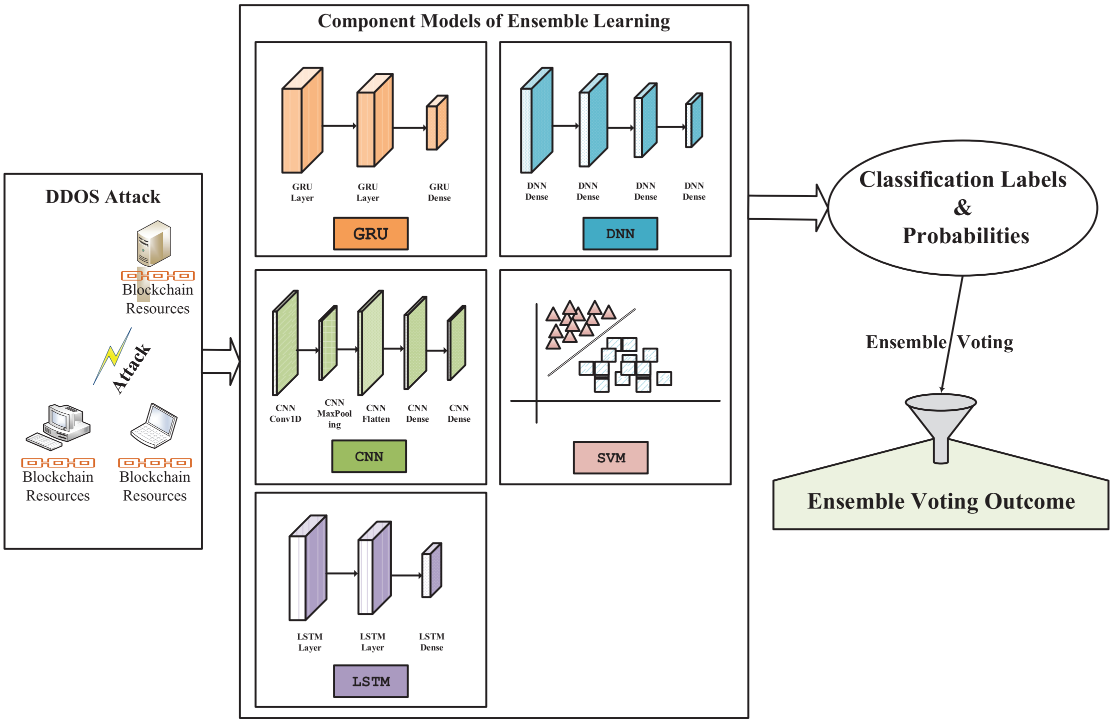

Figure 3: DDoS attack detection architecture diagram

In the event of a network attack, each node autonomously executes its respective models to evaluate the condition of the network. The rigorous voting procedure entails a straightforward application of the majority rule, wherein every node casts a vote to determine the occurrence of an attack. In the event that a significant number of nodes collectively ascertain the presence of an attack, the entire network will achieve a consensus and consequently conclude that an attack is truly taking place. This methodology leverages the decentralized characteristics of blockchain technology to enhance the reliability of detection. Additionally, it augments the confidence level of detection outcomes by employing a collective decision-making technique. The implementation of an autonomous assessment of each node, followed by a rigorous voting mechanism, enhances the precision and resilience in the detection and mitigation of DDoS assaults.

It is important to acknowledge that we also take into account the temporal aspects of DDoS traffic data. Specifically, we organize the data using sliding window time slices and compute a set of weights for each time slice. In each batch, there are 150 test samples, and the model initially computes the prediction accuracy of the individual submodels. If the accuracy of the forecast surpasses a predefined threshold, such as 0.8, the weight assigned to the submodel remains unaltered. Conversely, if the accuracy falls below the threshold, the weight assigned to the submodel is adjusted to zero. The weights are subsequently normalized and employed for weighted soft voting in the present batch. Subsequently, these weights are employed in a weighted averaging process, enabling the model to adjust to the varying attributes that may be present in the data across different time intervals.

Ultimately, the weights of each individual batch are combined by averaging to yield a global weight vector. In order to enhance the performance of the model, an objective function is formulated that utilizes negative accuracy as a loss metric. This choice is motivated by the aim to maximize the accuracy of the model. The constraint guarantees that the total sum of weights equals 1. After undergoing several iterations, an optimized set of weights is obtained for the ultimate weighted soft vote. This facilitates the achievement of more precise categorization in various situations. This collection of approaches integrates the benefits of hard voting and soft voting while also incorporating considerations for time dependency and model confidence. As a result, it achieves an effective and precise system for detecting DDoS attacks, demonstrating both efficiency and accuracy.

As shown in Fig. 4, we divide the data into segments after arranging them according to the time series, as shown in the figure, we divide them into N segments, representing N consecutive moments, and at each moment, we can get a classification result according to $N = 5$ models, expressed by the formula:

$$\hat{y}_{\mathrm{ensemble\_k}} = argmax_j \sum_{i=1}^{N} w_{i\_k} \times y_{i\_l} \tag{18}$$

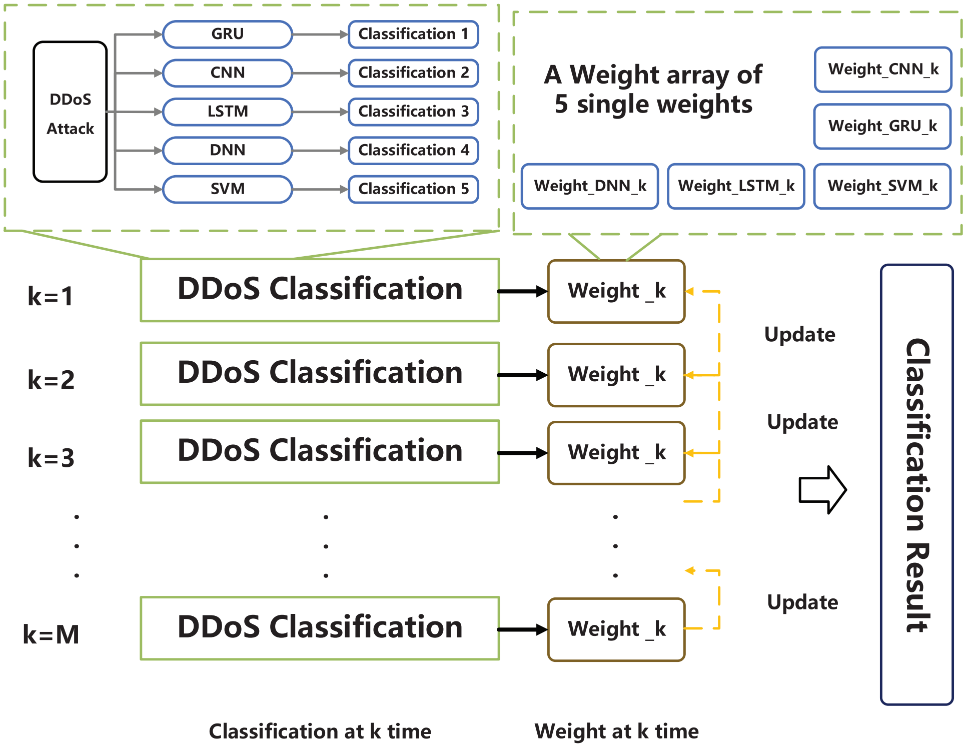

Figure 4: DDoS attack classification architecture diagram

$\hat{y}_{\mathrm{ensemble\_k}}$ represents the prediction result of the ensemble model at time k. w_{i_k} represents the weight of the ith basic classifier. *Weight_k* is a group of the w_{i_k}. At the first moment ($k = 1$), we define the model weight for each model to 0.2. y_{i_l} represents the result that the ith basic classifier predicts the DDoS attack as a class l, which it refers to the 7 types according to the Section 4.1.

$k > 2$, we first get the prediction accuracy of each individual model at that moment k, which we define as R_{i_k}, we select the model that R_{i_k} greater than the correct detection threshold, and set the others to 0, that is, do not participate in the fusion of weights. It can be expressed as:

$$Weight_p\_k = \frac{1}{\sum_{i=1}^{N} R_{i\_k}} [R_{1\_k} R_{2\_k} R_{3\_k} R_{4\_k} R_{5\_k}] \tag{19}$$

We apply the sliding window for adaptive allocation of weights, we choose the sliding window width as d, then

$$Weight\_M = \frac{1}{d+1} \sum_{k=M-d}^{k=M} Weight\_k + Weight_p\_M \tag{20}$$

The advantage of this is that the iteration weights can be continuously updated in the time series, which not only contains the forecast information of each model at this moment. It also contains model information for the previous time period.

5 Simulation and Analysis

5.1 Detection Results and Analysis

The initial step involves loading the preprocessed dataset and subsequently saving the extracted features and corresponding labels. The characteristics are subsequently normalized using the MinMaxScaler technique in order to guarantee that they fall within a consistent range of values. Next, the dataset is partitioned into a training set and a test set using the train_test_split function, with the test set comprising 20% of the total dataset. The model architecture for each model, namely GRU, CNN, LSTM, DNN, and SVM, is specified, and the relevant parameters are configured. Subsequently, the models are trained, and the performance metrics are monitored and recorded during the training process. Subsequently, the model that has undergone training is employed to generate predictions on the test dataset and subsequently evaluate its performance by computing metrics such as accuracy, precision, recall, and F1 score. The Results DataFrame contains the performance results and names of each model. Furthermore, a class called KerasClassifierWrapper was implemented to encapsulate the Keras model into a model that is compatible with the sklearn interface, specifically designed for constructing hard-voting ensemble models. A soft voting ensemble approach is generated by doing calculations to determine the anticipated probability of the model. Ultimately, the performance metrics of the ensemble model are computed. The performance data of all models and method names are consolidated into a DataFrame. The performance data of each individual model and ensemble method is then presented through the printed output.

We analyze using common evaluation metrics: Accuracy, Precision, Recall, and F1 score.

(1) Accuracy: The ratio of the number of correct predictions to the number of all predictions.

(2) Precision: The ratio of positive cases predicted correctly to all positive examples predicted positively.

(3) Recall: The ratio of correctly predicted positive examples to all examples that are actually positive.

(4) F1 score: Harmonic averages of Precision and Recall.

As shown in Table 1, during the DDoS attack detection evaluation (refer to Table 1), we thoroughly assessed the performance of various models. DNNs stood out, achieving the highest accuracy, precision, and F1 scores—around 0.9988, 0.9981, and 0.9988, respectively. The DNN showcased remarkable abilities in accurately identifying DDoS attack traffic. Additionally, CNN and SVM performed exceptionally well, reaching accuracies of approximately 0.9942 and 0.9945, recall values of around 0.9928 and 0.9933, and F1 scores of about 0.9942 and 0.9945, respectively. This suggests that both CNNs and SVMs are promising in detecting and distinguishing DDoS attack patterns, comparable to DNNs, and provide satisfactory results.

Table 1: Detection performance comparison table

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
GRU	86.57	81.59	94.48	87.56
CNN	99.42	99.57	99.28	99.42
LSTM	95.07	95.88	94.19	95.03
DNN	99.88	99.81	99.95	99.88
SVM	99.45	99.57	99.33	99.45
Hard vote	99.50	99.57	99.42	99.50
Soft vote	99.71	99.76	99.66	99.71

Nevertheless, the performance of models belonging to the RNN category, such as the GRU and LSTM, was comparatively subpar in the aforementioned evaluation. The GRU model achieved an accuracy of 0.8657, a precision of 0.8159, a recall of 0.9448, and an F1 score of 0.8756. On the other hand, the LSTM model achieved an accuracy of 0.9507, a precision of 0.9588, a recall of 0.9419, and an F1 score of 0.9503. One possible explanation for this phenomenon is that RNN models exhibit inferior performance compared to alternative models when confronted with network traffic characterized by high dimensionality and intricate relationship patterns, hence constraining their overall effectiveness.

In general, the multi-model adaptive technique that we have suggested has highly satisfactory performance. This strategy effectively allocates weights to each model in order to accommodate various data scenarios, hence enhancing their complementarity in diverse settings. The efficacy of this approach demonstrates that the integration of many models can greatly enhance the precision and dependability of DDoS attack detection, hence offering a robust safeguard for network security.

5.2 Classification Results and Analysis

In this part, in order to address the intricate issue of detecting DDoS attacks, we made the decision to incorporate a combination of several models. The underlying principle of this technique is rooted in the notion that distinct machine learning models possess distinct strengths when it comes to handling data with varying features. The integration of numerous models allows for the utilization of the individual strengths possessed by each model, thereby mitigating their respective limitations and ultimately enhancing the total accuracy of detection. Each model, namely GRU, CNN, LSTM, DNN, and SVM, was initially delineated, subsequently trained to utilize the training data, denoted as X_train and y_train. During this procedure, GRU, CNN, and LSTM required the data to be transformed into three-dimensional structures, while DNNs and SVMs employed the raw data without any alterations. Following the training of each model, a subsequent evaluation was conducted using test data (X_test and y_test). Performance metrics, including accuracy, precision, recall, and F1 score, were computed and subsequently stored in the respective DataFrame (e.g., results_gru, results_cnn, etc.). Furthermore, for the purpose of model integration, the final prediction outcome is determined through a process of hard voting, where the majority category is selected based on the prediction results of each individual model. Optimization algorithms are employed to allocate weights to individual models in order to get optimal ensemble outcomes. The optimized soft voting ensemble model utilizes cross-validation and minimization of the objective function to determine the optimal weights. These weights are then employed to compute the prediction outcomes.

This paper thoroughly analyzes various models for classifying DDoS attacks into seven categories (shown in Table 2). The GRU model exhibits somewhat subpar performance, with an accuracy and recall of approximately 0.7609, and an F1 score of about 0.7212. In contrast, the CNN model performs notably well, achieving an accuracy and recall of around 0.8590 and an F1 score close to 0.8583, indicating effective categorization of DDoS attack traffic. The LSTM model demonstrates average performance, with an accuracy and recall of roughly 0.8109 and an F1 score near 0.8090. The DNN model stands out with an accuracy and recall of approximately 0.8614 and an F1 score around 0.8602. The SVM model shows satisfactory but slightly less remarkable performance, achieving accuracy and recall values of about 0.8500 and an F1 score of roughly 0.8492. As for ensemble methods, they enhance performance, with soft voting achieving accuracy and recall of around 0.8680 and an F1 score of about 0.8670. The Adaptive Weight Method further optimizes results, posting values close to 0.8762 for accuracy and recall and an F1 score of approximately 0.8756.

Table 2: Classification performance comparison table

Method	Accuracy (%)	Precision (%)	Recall (%)	F1 score (%)
GRU	76.09	74.35	76.09	72.12
CNN	85.90	86.18	85.90	85.83
LSTM	81.09	81.43	81.09	80.90
DNN	86.14	86.48	86.14	86.02
SVM	85.00	85.58	85.00	84.92
Hard vote	85.57	85.86	85.57	85.46
Soft vote	86.80	86.98	86.80	86.70
Adaptive weight method	87.62	87.70	87.61	87.56

Fig. 5 provides a comprehensive comparison of performance metrics for DDoS classification models, highlighting the robust performance of the DNN model. This representation further emphasizes the superior efficacy of ensemble techniques and the outcomes achieved through optimization methodologies. The model's notable accuracy, precision, and F1 score serve as evidence of its supremacy. Both CNN and SVM models demonstrate high performance, often comparable to or even surpassing the effects achieved by DNN models. Nevertheless, the performance of the GRU and LSTM models in this particular test was found to be subpar, potentially attributable to their inherent constraints in effectively capturing intricate interactions. Ultimately, the multi-model adaptive method successfully attains desirable outcomes in terms of performance, hence substantiating the efficacy of multi-model fusion in the context of DDoS attack detection.

Ensemble models exhibit superior capability in managing data diversity and dynamics compared to conventional single-model methodologies, including GRU, CNN, LSTM, DNN, or SVM in isolation. The utilization of this integrated strategy demonstrates enhanced stability and precision when confronted with diverse types of DDoS attacks. Based on the aforementioned findings, it is evident that models such as DNN, CNN, and SVM exhibit commendable performance in terms of detection accuracy. However, it is important to acknowledge that these models may encounter constraints while processing certain data features. By incorporating these models into our framework, we can optimize performance across many scenarios and environments.

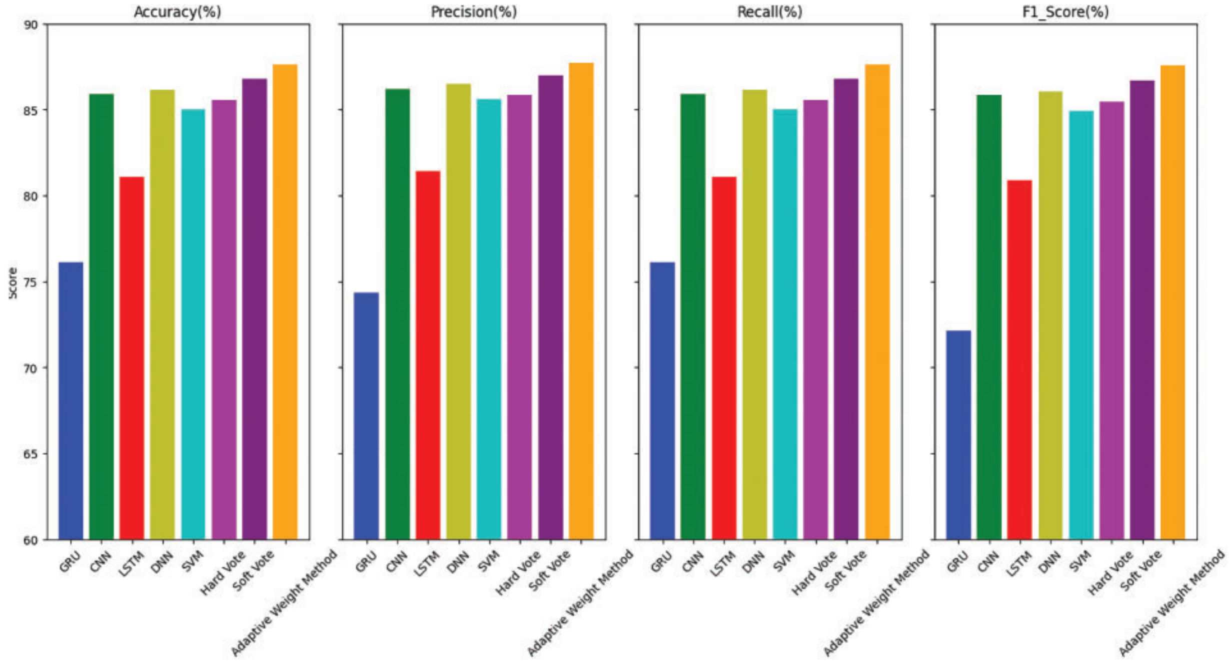

Figure 5: Performance metrics of models of DDoS classification

5.3 *Comparison with Previous Studies*

In the endeavor to identify the most effective methods for DDoS attack detection, it is pivotal to juxtapose contemporary techniques against preceding methodologies. As delineated in Table 3, the proposed method of this study outpaces several other widely-used techniques in terms of classification accuracy. With an impressive accuracy of 99.71%, our method not only exhibits superior performance but also supports segmentation with an efficiency of 87.62%. This is particularly noteworthy considering that certain recognized models, such as Light Gradient Boosting Machine (LightGBM) [41] and SVM [42] do not extend support for segmentation.

Table 3: Comparative analysis of DDoS attack detection methods

Attribute	Detection accuracy	Classification accuracy	Characteristics
Our method	**99.71%**	**87.62%**	It seamlessly merges five models within blockchain nodes, ensuring decentralized DDoS discernment through soft voting.

(Continued)

Table 3 (continued)

Attribute	Detection accuracy	Classification accuracy	Characteristics
LightGBM [41]	99.56%	Not support	It operates with minimal computational overhead while maintaining high precision.
SVM [42]	99.41%	Not support	It adeptly performs real-time anomaly-centric intrusion detection activities.
Naïve Bayes (NB) [43]	95.14%	Not support	It efficiently predicts classifications for the test datasets in a swift manner.

The inherent strength of our method lies in its unique architectural framework—an ensemble of five models incorporated seamlessly within blockchain nodes, harnessing the potential of decentralized processing. This soft voting technique ensures amplified accuracy, making it a formidable tool in the DDoS detection arsenal. On the contrary, while LightGBM [41] and SVM [42] boast commendable classification accuracies of 99.56% and 99.41%, respectively, they lack the capability to handle segmentation tasks. LightGBM [41] stands out for its economical computational requirements despite ensuring high precision, and SVM [42] is laudable for its real-time anomaly-based intrusion detection proficiency. In contrast, the Naïve Bayes (NB) [43] algorithm, with its expedient prediction capabilities for test datasets, offers a classification accuracy of 95.14%.

In this research, we introduced a novel multi-model ensemble technique, drawing its strength from decentralization and autonomy, which substantially surpasses traditional methods in DDoS attack detection. Traditional centralized detection strategies, which inherently rely on a singular point of reference, are often riddled with vulnerabilities. In stark contrast, our approach harnesses the robust nature of decentralized networks. By employing this innovative structure, every node in the system operates with unparalleled autonomy, significantly reducing vulnerabilities typically associated with a singular point of reference.

The ensemble approach adopted is particularly noteworthy. Different models under this ensemble have unique strengths and capabilities. For instance, while deep neural networks excel in the precise identification of DDoS attack traffic, convolutional neural networks, and support vector machines have their distinct advantages. By seamlessly integrating the strengths of these diverse models, our system ensures a holistic coverage of potential attack patterns. Moreover, ensemble techniques in the realm of machine learning have empirically proven their mettle in enhancing generalization, curtailing biases, and providing a robust defense against overfitting.

When compared to some previous methodologies that have leaned heavily on singular detection models, our approach emerges as far more adaptive. Singular models, though effective to a degree, often struggle when faced with rapidly evolving attack patterns. This adaptability challenge is what our multi-model ensemble squarely addresses. Additionally, a key drawback of conventional methods lies in their centralized structure. Such systems, due to their inherent design, grapple with scalability constraints and can become easy targets. Our approach, by juxtaposing the detection mechanism within the decentralized fabric of networks, not only sidesteps these challenges but also ensures the system's swift adaptability to novel threats, safeguarding networks from a diverse array of evolving threats with minimal recalibration.

6 Conclusion

In this study, we introduced a groundbreaking ensemble learning strategy that amalgamates the strengths of five diverse machine learning models: GRU, CNN, LSTM, DNN, and SVM. This fusion aims to counteract the intricate patterns of DDoS assaults more effectively than any individual model can. The superiority of this integrated approach, especially in terms of accuracy and speed, becomes evident when confronted with the complex and continuously evolving nature of data traffic, especially within blockchain networks. While the dynamic weight adjustment technique significantly bolsters our approach by curbing overfitting, it simultaneously augments the generalization prowess of the model. However, a comprehensive perspective demands the recognition of inherent limitations. The dataset utilized, although extensive, may not fully encapsulate the complexities of real-world network traffic, thus potentially influencing the model's real-world generalization and precision. Moreover, as our methodology leans on multiple models, it brings forth challenges in elucidating the combined model's behavior, especially in practical applications. Future avenues for this research are multifold. We aim to delve deeper into refining detection methodologies tailored specifically for distinct attack patterns in blockchain contexts. Considering the integration of more models and sophisticated algorithms is also on the horizon, aiming to further reinforce the robustness and accuracy of our system. Additionally, the pivotal concern of model interpretability, which remains a frontier in machine learning, will form a cornerstone of our subsequent endeavors. In summation, this research underscores the profound importance of enhancing the defense mechanisms of blockchain technologies. By persistently pushing the boundaries through innovative strategies, we endeavor to make invaluable contributions to fortifying the security landscape of blockchain technology.

Acknowledgement: We thank Hainan University's Computer Science and Cyberspace Security Schools and the Hainan Blockchain Technology Center for their resources and support. Our gratitude goes to Hainan Hairui Zhong Chuang Technol Co., Ltd., for their collaboration. Special thanks to Qiongtai Normal University and Mengzhe Sun for their contributions.

Funding Statement: This work was supported by National Natural Science Foundation of China (Grant Nos. 62162022 and 62162024), Hainan Provincial Natural Science Foundation of China (Grant Nos. 723QN238 and 621RC612).

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: X. Li, J. Cheng; data collection: C. Ruan, B. Zhang, M. Sun; analysis and interpretation of results: X. Li, J. Cheng, X. Tang; draft manuscript preparation: X. Tang, M. Sun. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The data is available on https://www.unb.ca/cic/datasets/ddos-2019.html.

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

References

[1] Y. Lu, "The blockchain: State-of-the-art and research challenges," *Journal of Industrial Information Integration*, vol. 15, no. 3, pp. 80–90, 2019.

[2] Y. Lu, "Blockchain and the related issues: A review of current research topics," *Journal of Management Analytics*, vol. 5, no. 4, pp. 231–255, 2018.

[3] W. Zhao, "Blockchain technology: Development and prospects," *National Science Review*, vol. 6, no. 2, pp. 369–373, 2019.

[4] X. Zheng and Y. Lu, "Blockchain technology—Recent research and future trend," *Enterprise Information Systems*, vol. 16, no. 12, pp. 1939895, 2022.

[5] G. B. Mermer, E. Zeydan and S. S. Arslan, "An overview of blockchain technologies: Principles, opportunities and challenges," in *2018 26th Signal Processing and Communications Applications Conf.*, Izmir, SIU, Turkey, pp. 1–4, 2018.

[6] R. Paulavičius, S. Grigaitis, A. Igumenov and E. Filatovas, "A decade of blockchain: Review of the current status, challenges, and future directions," *Informatica*, vol. 30, no. 4, pp. 729–748, 2019.

[7] D. V. Lypnytskyi, "Opportunities and challenges of blockchain in Industry 4.0," *Economy of Industry*, vol. 1, no. 85, pp. 82–100, 2019.

[8] R. Singh, S. Tanwar and T. P. Sharma, "Utilization of blockchain for mitigating the distributed denial of service attacks," *Security and Privacy*, vol. 3, no. 3, pp. 96, 2020.

[9] S. Wani, M. Imthiyas, H. Almohamedh, K. M. Alhamed, S. Almotairi *et al.,* "Distributed denial of service (DDoS) mitigation using blockchain—A comprehensive insight," *Symmetry*, vol. 13, no. 2, pp. 227, 2021.

[10] 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.

[11] R. Santos, D. Souza, W. Santo, A. Ribeiro and E. Moreno, "Machine learning algorithms to detect DDoS attacks in SDN," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 16, pp. 5402, 2020.

[12] D. A. Banitalebi, M. Soltanaghaei and F. Z. Boroujeni, "The DDoS attacks detection through machine learning and statistical methods in SDN," *The Journal of Supercomputing*, vol. 77, no. 1, pp. 2383–2415, 2021.

[13] M. N. Faiz, O. Somantri, A. R. Supriyono and A. W. Muhammad, "Impact of feature selection methods on machine learning-based for detecting DDoS attacks: Literature review," *Journal of Informatics and Telecommunication Engineering*, vol. 5, no. 2, pp. 305–314, 2022.

[14] K. Bouzoubaa, Y. Taher and B. Nsiri, "Predicting DOS-DDOS attacks: Review and evaluation study of feature selection methods based on wrapper process," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 5, pp. 131–145, 2021.

[15] A. A. Bahashwan, M. Anbar, S. Manickam, T. A. Al-Amiedy, M. A. Aladaileh *et al.,* "A systematic literature review on machine learning and deep learning approaches for detecting DDoS attacks in software-defined networking," *Sensors*, vol. 23, no. 9, pp. 4441, 2023.

[16] M. Najafimehr, S. Zarifzadeh and S. Mostafavi, "DDoS attacks and machine-learning-based detection methods: A survey and taxonomy," *Engineering Reports*, vol. 1, no. 1, pp. 12697, 2023.

[17] X. Yuan, C. Li and X. Li, "DeepDefense: Identifying DDoS attack via deep learning," in *2017 IEEE Int. Conf. on Smart Computing*, Hong Kong, China, pp. 1–8, 2017.

[18] S. Sambangi and L. Gondi, "A machine learning approach for ddos (distributed denial of service) attack detection using multiple linear regression," *Proceedings*, vol. 63, no. 1, pp. 51, 2020.

[19] T. Liu, F. Sabrina, J. Jang-Jaccard, W. Xu and Y. Wei, "Artificial intelligence-enabled DDoS detection for blockchain-based smart transport systems," *Sensors*, vol. 22, no. 1, pp. 32, 2021.

[20] Z. Yin, K. Li and H. Bi, "Trusted multi-domain DDoS detection based on federated learning," *Sensors*, vol. 22, no. 20, pp. 7753, 2022.

[21] R. Jmal, W. Ghabri, R. Guesmi, B. M. Alshammari, A. S. Alshammari *et al.,* "Distributed blockchain-SDN secure IoT system based on ANN to mitigate DDoS attacks," *Applied Sciences*, vol. 13, no. 8, pp. 4953, 2023.

[22] Z. Liu and X. Yin, “LSTM-CGAN: Towards generating low-rate DDoS adversarial samples for blockchain-based wireless network detection models,” *IEEE Access*, vol. 9, pp. 22616–22625, 2021.

[23] H. Mrabet, A. Alhomoud, A. Jemai and D. Trentesaux, “A secured Industrial Internet-of-Things architecture based on blockchain technology and machine learning for sensor access control systems in smart manufacturing,” *Applied Sciences*, vol. 12, no. 9, pp. 4641, 2022.

[24] E. S. Babu, B. K. N. Srinivasa Rao, S. R. Nayak, A. Verma, F. Alqahtani *et al.,* “Blockchain-based intrusion detection system of IoT urban data with device authentication against DDoS attacks,” *Computers and Electrical Engineering*, vol. 103, no. 1, pp. 108287, 2022.

[25] B. Ilyas, A. Kumar, M. A. Setitra, Z. A. Bensalem and H. Lei, “Prevention of DDoS attacks using an optimized deep learning approach in blockchain technology,” *Transactions on Emerging Telecommunications Technologies*, vol. 34, no. 4, pp. 4729, 2023.

[26] C. Xu, Y. Qu, T. H. Luan, P. W. Eklund, Y. Xiang *et al.,* “An efficient and reliable asynchronous federated learning scheme for smart public transportation,” *IEEE Transactions on Vehicular Technology*, vol. 72, no. 5, pp. 6584–6598, 2022.

[27] R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, S. Garg *et al.,* “A distributed intrusion detection system to detect DDoS attacks in blockchain-enabled IoT network,” *Journal of Parallel and Distributed Computing*, vol. 164, no. 1, pp. 55–68, 2022.

[28] H. Huang, P. Ye, M. Hu and J. Wu, “A multi-point collaborative DDoS defense mechanism for IIoT environment,” *Digital Communications and Networks*, vol. 9, no. 2, pp. 590–601, 2023.

[29] A. Rojarath and W. Songpan, “Probability-weighted voting ensemble learning for classification model,” *Journal of Advances in Information Technology*, vol. 11, no. 4, pp. 217–227, 2020.

[30] A. Ekinci and H. I. Erdal, “Forecasting bank failure: Base learners, ensembles and hybrid ensembles,” *Computational Economics*, vol. 49, no. 4, pp. 677–686, 2017.

[31] Y. Guo, X. Wang, P. Xiao and X. Xu, “An ensemble learning framework for convolutional neural network based on multiple classifiers,” *Soft Computing*, vol. 24, no. 1, pp. 3727–3735, 2020.

[32] M. Aslam, D. Ye, A. Tariq, M. Asad, M. Hanif *et al.,* “Adaptive machine learning based distributed denial-of-services attacks detection and mitigation system for SDN-enabled IoT,” *Sensors*, vol. 22, no. 7, pp. 2697, 2022.

[33] 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, USA, pp. 45–59, 2016.

[34] 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*, Istanbul, Turkey, pp. 190–199, 2021.

[35] J. Cheng, X. Li, X. Xu, X. Tang and V. S. Sheng, “A modified PointNet-based DDoS attack classification and segmentation in blockchain,” *Computer Systems Science & Engineering*, vol. 47, no. 1, pp. 975–992, 2023.

[36] M. Subrmanian, K. Shanmugavadivel, P. S. Nandhiniand and R. Sowmya, “Evaluating the performance of LSTM and GRU in detection of distributed denial of service attacks using CICDDoS2019 dataset,” in *7th Int. Conf. on Harmony Search, Soft Computing and Applications: ICHSA 2022*, Singapore, pp. 395–406, 2022.

[37] Z. X. Yang, X. L. Qin, W. R. Li and Y. J. Yang, “A DDoS detection approach based on CNN in cloud computing,” *Applied Mechanics and Materials*, vol. 513, no. 1, pp. 579–584, 2014.

[38] B. Wang, P. Wang and Y. Tu, “Customer satisfaction service match and service quality-based blockchain cloud manufacturing,” *International Journal of Production Economics*, vol. 240, no. 1, pp. 108220, 2021.

[39] J. Zhao, M. Xu, Y. Chen and G. Xu, “A DNN architecture generation method for DDoS detection via genetic alogrithm,” *Future Internet*, vol. 15, no. 4, pp. 122, 2023.

[40] Z. Ma and B. Li, “A DDoS attack detection method based on SVM and K-nearest neighbour in SDN environment,” *International Journal of Computational Science and Engineering*, vol. 23, no. 3, pp. 224–234, 2020.

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

[42] 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.

[43] 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.