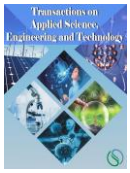
Lanash
Transactions on Applied Science, Engineering and Technology
Research Paper
Volume 2 Issue 2
Published 05 October 2026



The Role of Blockchain and AI-Based Cyber Security for the Detection and Protection of DoS Attacks

E.T. Mitiku*[1], T. Beshah[2], H. Kassaw[1] and A. Endalew[3]

[1]Electrical and Computer Engineering, University of Gondar, Gondar, Ethiopia
[2]Information Systems, Addis Ababa University, Addis Ababa, Ethiopia
[3]Digital Transformation and Innovation, University of Gondar, Gondar, Ethiopia

Submitted on 25 August 2026
Accepted on 17 September 2026
Published on 05 October 2026

To cite this article: E.T. Mitiku, T. Beshah, H. Kassaw and A. Endalew, "The Role of Blockchain and AI-Based Cyber Security for the Detection and Protection of DoS Attacks," *Trans. Appl. Sci. Eng. Technol.*, vol. 2, no. 2, pp. 1-11, 2026.

Copyright:

Abstract
At this time, network systems are facing the challenges of so many cyber-attacks. DoS attack needs a special attention and techniques so as to make these systems free from different cyber-attacks. There are emerging technologies for the protection of cyber-criminals. For example, Blockchain technology has vital features for fighting attacks. This study is formulated for the prevention of denial-of-service attacks. Selected methodologies are used to fulfil the goal of the proposed study. Appropriate models are trained on the relevant dataset. The CNN model has the best values of 98.3% and 100% detection accuracy and prevention ability, respectively.

Keywords: AI and blockchain technology; DoS attack; detection; protection

1. Introduction

In the current century, Emergent technologies can be defined as those that is commonly utilized to be described as modern technologies, and the evolution of existing technology. It is widely used to describe technologies that are in development or will be available within the next few years. It is usually reserved for technologies that are creating or are expected to create significant social or economic effects. When employed in different contexts, it has different contextual meanings depending on the applied area or field. Emerging Technology includes data science, artificial intelligence (AI), virtual reality, and augmented reality, Big Data, cloud computing, fog computing, and connection-oriented electronic devices and Blockchain Technology [7]. In the recent time, the fastest modern exchange of data growth resulted an increase of both internet-based service and application which make the client-server architecture the focal point of digitalization. But with this a purposeful progress of client-server network, there is a cyber-Security challenge which is called Denial of Service (DoS) cyberattacks. It does overpower a client-server network architecture with a flood of request traffic from many sources, denying lawful users the ability to get service from the server. This DoS attack causes not only disruption of operation, financial loss, and distraction of any reputation. The common cybersecurity methods of client-server-based networks, for example, firewalls, antivirus, intrusion detection systems (IDS), Artificial Intelligence (AI), Machine Learning (ML), and Deep Learning (DL) not have the ability to minimize or eliminate DoS cybersecurity challenges. As a result, this thesis is proposed using a Blockchain Technology-based cybersecurity framework as Blockchain Technology has a decentralized and immutable nature, and it brings a great choice for client-server networks, which withstands DoS attacks because it improves the cyberspace. No organization can predict when a DoS attack will occur. However, malicious actors often look for gaps in security systems to launch a DoS attack [8]. Blockchain has been considered a modern solution to the current cybercrime activities. There is a transparent condition among each participant node; since each node shares a common storage called the shared ledger, and there are distributed characteristics. Every record on that storage, and once a transaction record is stored, the record is going to be immutable. These properties lead the Blockchain technology to be selected as a great protector of Denial-of-Service Attack problems. Blockchain-based cybersecurity systems have automatic cybercrime prevention capability using Blockchain smart contracts that work based on pre-programmed rules and regulations [20-22], [24-25], [28-29], and [44]. The continuous cybercrimes in the AI-based system create the worst problems for the recent day's modern cyberspace. The traditional systems are always dependent on centralized systems, which have insufficient performance to prevent and reveal DoS attacks and modern cybersecurity challenges. The traditional systems of artificial intelligence (AI), machine learning (ML), and Deep Learning (DL) are limited only to the task of detecting DoS attacks; they are not able to protect the client-server network cyberspace from DoS attacks. Centralized-based property of the traditional systems produces a particular failure that makes them more attractive targets in cyberspace for DoS attack criminals. The difficulty of deploying Artificial Intelligence (AI) cybersecurity techniques often results in poor DoS attack protection performance, leaving organizations vulnerable and susceptible to DoS attackers [52] and [54].

1.1. Statement of the problem

Greene et al. proposed a study using user interaction, network traffic management, server node verification, and mining as research techniques. The study found that the system server was unable to serve clients properly because it was overloaded. There were flaws in the research, such as a lower scale, a centralized environment, a narrow DoS attack scope, and no prevention testing. Hussain et al. conducted this research, which has a misclassification; SYN and DNS, which are detected 100% precision, and UDP is misclassified as NetBIOS, requiring high computational resources, etc. Models are dependent only on one dataset. In this research, methods like data

*Corresponding Author:
E.T. Mitiku, Electrical and Computer Engineering, University of Gondar, Gondar, Ethiopia

collection, data preprocessing, model training, and attack detection [3]. Alabdulatif et al. assured the study that focused on a single dataset, and there is only one task, which is detection. ML models (XGB, RF, and MLP), feature engineering, and prototype are all techniques used in this study [4]. Coscia et al. participated in a study that had a low protocol scope and data quality independent. The method used in this study is Anomaly2Detection to get the best classification of 99.97% to 99.99% [48].

This thesis is designed with the help of these research questions:

- Which models are the best for DoS attack detection?
- What is the gap in current DoS-based cyber-attack reducing techniques?
- What are the parameters to check the effectiveness of the proposed study?
- How can the proposed blockchain technology-based model be supported to protect DoS attacks?

1.2. Objectives of the study

1.2.1. General objectives of the study

The major goal of this study is to formulate a strong Blockchain -based cyber Security technique for detecting and protecting against DoS attacks.

1.2.2. Specific objectives of the study

- To select the best model for DoS attack detection.
- To identify the gap in the current Denial of Service reduction techniques.
- To make a performance evaluation testing using parameters to check the proposed Blockchain-based Cyber Security model's efficiency.
- To propose a blockchain-based model to protect the DoS attacks?

2. Related Works

Greene et al. announced that the traditional systems are always dependent on centralized systems, which have insufficient performance to prevent and reveal DoS attacks and modern cybersecurity challenges. The common techniques of Machine Learning (ML), and Deep Learning (DL) are limited only to the task of detection of DoS attacks; they are not able to protect the client-server network cyberspace from DoS attacks. The centralized nature of the traditional systems produces a particular failure that makes them more attractive victims in cyberspace by DoS attack criminals. Some impacts of the so-called cyber-attack are financial loss, reputation damage, operational disruption and interruption, and increased costs on the client-server network architecture [2]. In the study of Hussain et al., an improved identification and reduction of DoS attack types has resulted, and there is also an improvement of cyber cybersecurity level and performance of the web server. A unification of traffic authentication into machine learning gives the power to protect against DoS attacks [3]. Alabdulatif et al. showed that integration of ML with new technologies has the highest ability for anomaly detection rates in a real-time way. ML is faster than traditional methods for DoS attack detection. There is a flexible nature for new skillful DoS attack identification for future DoS attacks due to MLs' prediction ability. The ability of ML to identify and protect against DoS attacks is dependent on the number of DoS attacks, and it cannot scan for false positives and false negatives [4]. Coscia et al. approved that a combination of ML techniques for automatic DoS detection and reactions results in an improved detection ability of DoS attacks, flexibility to new DoS attacks, an automated nature to monitor and control DoS attacks, and real-time reaction to DoS attacks. There is no flexible DoS attack detection and prevention technique for future adaptable and more skilled DoS attacks. ML has the ability to create false-positive and false-negative errors. The effectiveness of ML is dependent on the amount and quality of training data because ML's detection accuracy performance is affected by data quantity and data quality [48] **(Table 1)**.

3. Methodology

In this study, a number of techniques are used. For example, research review, dataset collection, dataset selection, dataset partition, model (ANN, CNN, KNN, LSTM, SVM, Random Forest, Hybrid (combination of LSTM and CNN) and Hybrid (combination of RF and XGBoost)) selection, classes imbalance handling technique, simulation and experimental, data analysis approaches, selection of performance evaluation parameters, model comparison and validation process; required hardware selection such as Hardware resources such as Data bus, Desktop, Ethernet cable, Hard Drive (HD), Personal Computer (PC), Secure Digital (SD) card, SIM card and Universal Serial Bus (USB) flash, and required software selection like CIC_DDoS2019 dataset, C++ programming languages, EndNote, Ganache Truffle tool, Ganache CLI, Git tool, HTML and CSS programming languages, JavaScript programming languages, Notepad ++, NS-3 Network Simulator, version-3.43 tool, Oracle VirtualBox for Kali Linux, PowerShell for Ubuntu, Python programming languages, Solidity programming languages, and Wondershare EdrawMax tool.

3.1. Designing of blockchain-based cyber security conceptual model framework

3.1.1. Unification of blockchain technology and normal network

It is a mixed architecture of the Blockchain technology network layer and the network layer.

A distributed ledger is created to hold client requests for the server information that are distributed among a set of participants. A request comes from any client, and the request is going to be authenticated and validated by the Blockchain network layer, and then, if it is a non-malicious request, it is passed into a Client-Server network layer server [54].

TABLE 1: Related works summary.

Author (year)	Title	Research Problem	Method	Finding	Limitation
Greene et al. (2018)	An Investigation into a Denial-of-Service Attack on an Ethereum Network	DoS attack on the Ethereum consortium networks	User interaction, Network traffic management, Server node verification, and mining	The sharp traffic increase overloaded the website and server, preventing request processing.	Lower scale, centralized environment, narrow DoS attack scope, no prevention testing
Hussain et al. (2020)	IoT DoS and DDoS Attack Detection using ResNet	Failure of Firewall and IDS to detect and prevent DoS and DDoS attacks. ResNet performs on raw network traffic data.	Data collection, Data preprocessing, model training, attack detection	Benign and attack classification:99.999% accuracy, multi-class classification: 87.06% accuracy, and 87% average precision,	Misclassification: SYN and DNS are detected 100% precision, and UDP is misclassified as NetBIOS. High computational resource needs, models are dependent on only one dataset.
Alabdulatif1 et al. (2023)	Machine Learning Enabled Novel Real-Time IoT Targeted DoS/DDoS Cyber Attack Detection System	DoS and DDoS attacks on IoT Networks.	ML models (XGB, RF and MLP), feature engineering and prototype.	Highest performance of XGB=99.93% accuracy, precision, recall and F1 score, real-time system, effective feature engineering validation	Single dataset, focus only on detection,
Coscia et al. (2024)	Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks	Cyber Security challenges on NIDIPS	Anomaly2Detection	Effective classification: 99.97% to 99.99%,	Low protocol scope and data quality independent.

3.1.1.1. Data flow through unification of blockchain and client-server network

Blockchain is located in between the client-server network layer and the source of the request and it acts as a filter and guard that keeps the request going to the client-server network layer section. A type of permissioned-based Blockchain technology is designed for the validation (approval) and filtration of any request from anywhere that must be validated, verified, and authenticated from the Blockchain network layer; after that, it must go to the client-server network layer service provider (server) to get the required response [54].

3.1.2. Blockchain-based cyber security conceptual model design

It describes more about the proposed Blockchain Technology based Cyber Security model and interactions of its components one another according to the collected data from literature reviews. Main parts of the conceptual Model of the Study are listed below [55]: Network Architecture, Threat Zone, and Blockchain Technology Layer.

3.2. Prototype implementation phase of the blockchain-based cyber security study

In the process of prototype implementation for the Blockchain based cyber security model, there are some number of crucial steps: Selection of Blockchain Platform=>Selection of IDE=>Selection of programming language=>Blockchain Integration=>Selection of network performance monitoring tool=>Selection of DoS attack generator tool/program=>Selection of Network Simulator tool=>Selection oof Statistical Analysis and Visualization tool=>Dev't of Smart Contract program=>Testing and Evaluating.

3.2.1. Denial of service attack detection program flow chart

The next flowchart shows the sequence of the simulation program execution process **(Figure 1)**. The SMOTE technique is incorporated into the simulation program to avoid if there is a class imbalance within a dataset, and so that it enables the simulation program to produce constant output through its repeated execution.

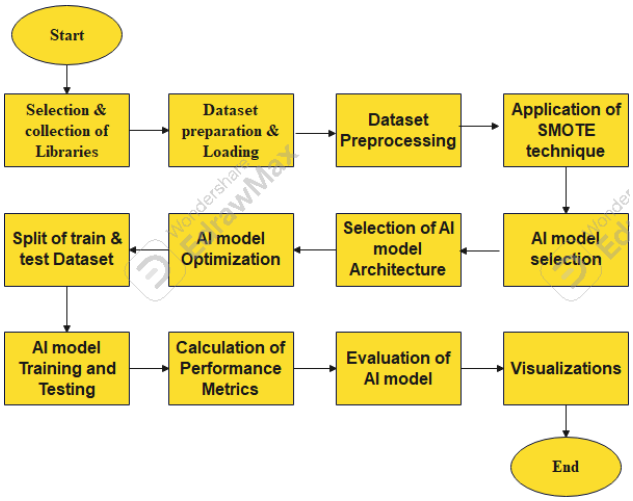


FIGURE 1: Prototype simulation program flowchart (Source: Wondershare EdrawMax).

3.2.2. Blockchain smart contract and DoS attack protection process

A smart contract is programmed and works based on a client request per second value (i.e., threshold=1000) that states if 1000 or above client requests come from a single client request IP address, we call it a DoS attack, is occurred, and then the IP address will be blocked according to [56] **(Figure 2)**.

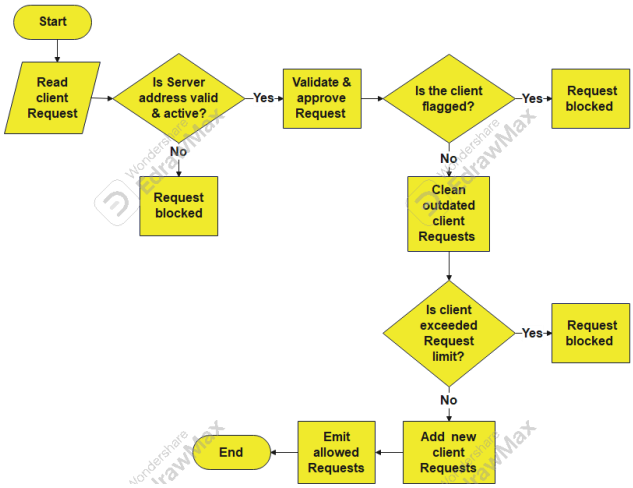


FIGURE 2: Smart contract DoS attack protection flowchart (Source: Wondershare EdrawMax).

4. Results and Discussions

4.1. Development and experimental environments

The CIC-DDoS2019 dataset is trained and tested using eight models to check the proposed study's effectiveness and validity.

4.1.1. Development and environmental setup preparation of DoS attack detection process

C++ programming language: It is used to develop a program for the preparation of NS-3 custom dataset (secondary or alternative dataset) on an IDE called Embarcadero Dev-C++ IDE.

Embarcadero Dev-C++ IDE: It is an IDE used to develop, run, edit and debug a C++ program for the preparation of NS-3 custom dataset (secondary or alternative dataset).

Google CoLab IDE: It is a cloud-based (virtual) IDE used to develop, run, edit and debug any program and to make a graphical output analysis.

Jupyter Notebook: It is a local IDE (offline) used develop, run, edit, and debug any program and to make a graphical output analysis.

Notepad++ Editor: It is a text editor used to develop or write, edit, copy, and cut a program.

NS-3: It is a simulator used to prepare a custom or secondary alternative type dataset (manual).

VSCode: It is a local IDE (offline) used to develop or write, run, edit, and debug any programming language.

The following table has values of performance evaluation metrics that are recorded during the prototype implementation phase of the simulation program **(Table 2)**.

Validation process: Let us take the output of the CNN model among the other models to explain the validity of our proposed Blockchain-Based Cyber Security, since it has the best DoS attack detection accuracy, DoS attack minimization, and protection capability. There are four classes, as shown in the following **(Figure 3)**, that have an AUC (Area Under Curve) value of 1, which indicates that there is complete (pure) categorization for individual classes using the CNN model.

Model Accuracy and Model Loss: At the left graph of the model accuracy section; training accuracy is symbolized by blue color and its value was approximately 0.92 and then increases to 1.00 by 100% when the training is completed which refers to the model (i.e CNN model) learnt on a CIC-DDoS2019 dataset. At this section of model accuracy; validation accuracy is colored by orange color and its value was approximately 0.18 and this value varied many times and that value was increased to 0.30 (constant). We can say that training accuracy is greater than validation accuracy (i.e 0.92 is greater than 0.18).

At the right graph of the model loss section; training loss is symbolized by blue color and its value was approximately 0.25 and then decreases to 0.00 when the training is completed that indicates the presence of higher value of training accuracy. At this section of model loss; validation loss is colored by range color and its value was approximately 5.4 and then it was varied many times and at the completion of the training phase; that value was decreased to 3.2 (constant). We can say that training loss is less than validation loss (i.e 0.00 is less than 3.2) **(Figure 4)**.

TABLE 2: Performance evaluation metrics of the continuous simulation record.

Latency (msec/samp)	Throughput (samp/sec)	Minimization and Prevention capability	Accuracy	AUC_ROC	CPU Utilization	Memory Utilization	Bandwidth Utilization	Model
0.85	748	93.8%	90.74%	93%	68.4%	82.7%	81.9MB	SVM
0.395	1285	99.7%	96.82%	98.74%	85.7%	90.0%	95.84MB	Hybrid
6.081	41.11	76%	99.24%	95.69%	81.7%	67.6%	3.5534MB	RF and XGBOOST
0.006	176608.13	99.42%	97.48%	91%	5.85%	73.70%	67.327MB	ANN
0.258	3871.29	92.01%	97%	87%	27.4%	80.3%	205.05 MB	KNN
0.011	92432.89	83.90%	99%	94%	25.0%	83.9%	463.50 MB	RF
380.169	14.11	100%	98.83%	97%	73.90%	2.70%	592.49 MB	CNN
0.079	12653.28	22.78%	78%	92.53%	67.5%	79.2%	438.78 MB	LSTM

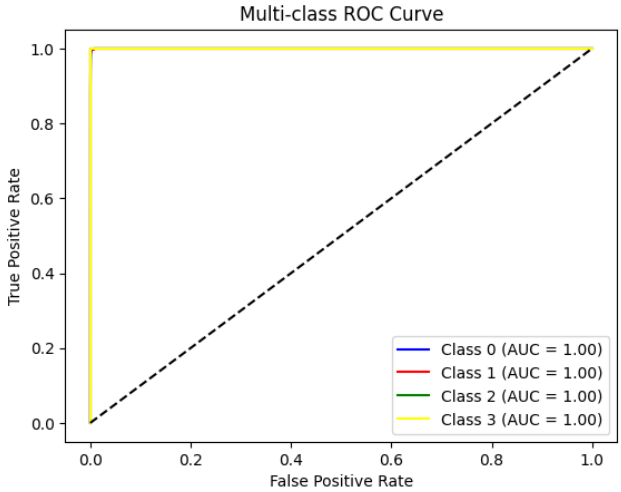


FIGURE 3: ROC curve.

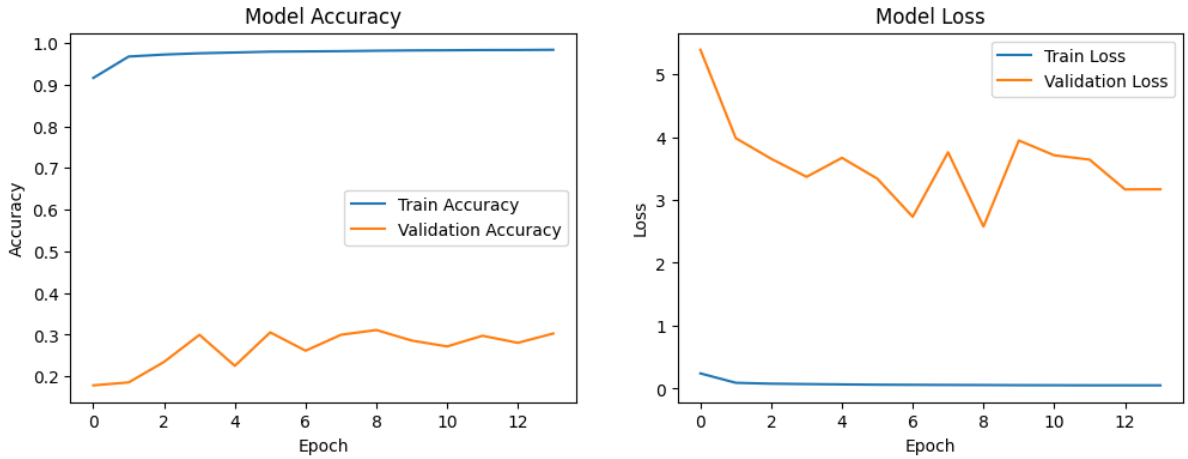


FIGURE 4: Model accuracy and loss.

Confusion Matrix: The confusion matrix is made up of four equal sized segments (i.e Square) that are represented as row (correct classes) and column (predictable classes). The Diagonal parts of the confusion matrix are called correct classes (TP) which are in the diagonal part and have the highest value and the off-diagonal parts of the confusion matrix are called misclassified classes that are numbers different from zero and above zero numbers outside matrix diagonal. Let us look at using an example based on the matrix. Let class 0 be correct class and numbers from 0 to 24 be predictable class indices.

From this confusion matrix, we can raise a conclusion like the diagonal part of the matrix is represented by TP; summation of matrix columns without diagonal is represented by FP; summation of matrix rows without diagonal is represented by FN and others are represented by TN and then we can calculate value of other metrics such as accuracy, precision, recall and F1 Score. The largest numerical values are located on the diagonal part of the confusion matrix, which is treated as the highest accuracy, and the other largest numerical values that are located outside the matrix diagonal are treated as FP **(Tables 3 and 4)**.

TABLE 3: Confusion matrix (CNN model).

Correct class	Predictable class	Explanations
0	[29083, 3, 1, 2, 2, 1, 1, 0, 0, 0, 3, 48, …]	The no. 29083 is categorized properly as it is class 0, but others are misplaced into other classes.
1	[78, 75, 0, 2, 0, 0, …]	No. 75 is categorized properly. But there are misplaced (78 is placed on class 0, and no. 2 is placed on class 3.)
2	[12, 1044, 0, 0, 0, 0, 42, …]	No. 1044 is categorized properly. 12 and 42 are misplaced as classes of 0 and 6, respectively.

TABLE 4: Confusion matrix (CNN model).

```
[[29083     3     1     2     2     1     1     0     0     0     3    48
     78    75     0     2     0     0]
 [   12  1044     0     0     0     0     0     0     0    42     0     0
      0     0     0     1     0     0]
 [    0     0   399     0     0     0     0     0     0     0     0     0
      0     0     0     0     0     0]
 [    1     0     0  1829     0     0     0     0     0     0     0     0
      0     0     0     0     0     0]
 [  123     0     0    14 36332     2     0     0     0     0     0     0
      0    17     0     0     0     0]
 [    0     0     0     0     0   155     0     0     0     0     0     0
      0     0     0     0     0     0]
 [    0     0     0     0     0     0   802     0     0     0     0     0
      0     0     0     0     0     0]
 [    2     0     0     0     0     0     0  3114     0     0     0     0
      0     1     0     0     0     0]
 [    0     0     0     0     0     0     0     0   553     4     0     0
      0     0     0     0     0     0]
 [    0    11     0     0     0     0     0     0    30  2466     0     3
      0    49     5     1     0     0]
 [    1     0     0     0     0     0     0     0     3     0   168     0
      0     1     1     0     0     0]
 [    4     0     0     0     0     0     0     0     0     0     0   191
      1     1     0     0     0     0]
 [   12     0     0     0     0     0     0     0     0     0     1     3
  14713    34     0     0     0     0]
 [   13     0     1     0     0     0     0     2     2     0     0     4
     17 29871     1     1     0     0]
 [    0   131     0     0     0     0     0     0     0    33    28     3
      2     0  5195     0     0     0]
 [   14     2     0     0     0     0     0     0     0     9     0     5
      0     1   618  1975     0     0]
 [    2     0     0     0     0     0     0     0     2     0     0     0
      3     1    12     0     0     0]
 [   16     0     0     0     0     0     0     0     0     0     0     0
```

4.1.1.1. Blockchain-based denial of service attack protection process

Simulation and development environmental setup preparation of DoS attack protection
It is a sequential process of program writing and simulating a program using a particular IDE. There are some important resources to accomplish this process.

Ganache CLI: It is used to emulate a local Ethereum blockchain network.

Ganache Truffle HD Wallet provider: It is a full package of fake accounts and private keys used to provide demo accounts and private keys, and to sign a transaction.

Google CoLab IDE: It is a cloud-based (virtual) IDE used to develop, run, edit, and debug any program and to make a graphical output analysis.

Jupyter Notebook: It is a local IDE (offline) used to develop, run, edit, and debug any program and to make a graphical output analysis.

MetaMask: It is a connection-based page used to assist a user in interacting with the Ethereum local network.

NPM terminal: It is a command-line terminal that is used to install and enable Ganache Truffle.

Remix IDE: It is an internet-based IDE used to develop or write, edit, debug and compile, and deploy Blockchain Smart contract programs.

Sepolia: It is a part of Metamask used to provide a testnet (test network) provider.

Solidity programming: It is a type of programming language used to develop Blockchain Smart contract programs.

VSCode: It is a local IDE (offline) used to develop or write, run, edit, and debug any program (i.e, truffle configuration file).

Web3.py: It is a new internet technology used to interact with a remote Ethereum local network.

The following tasks were done using crucial steps to prepare an interaction between the local Blockchain Ethereum networks.

MetaMask account preparation: Creating a MetaMask account and logging into it.

Blockchain Smart Contract program development: Writing a smart contract program using the Solidity programming language and cloud-based Remix IDE on the Ethereum blockchain platform.

Linkage between Remix and MetaMask: Preparation of an interaction between Remix IDE and MetaMask is essential to provide a testnet network provider from there for smart contract program deployment. On the Blockchain smart contract program (from Remix IDE), the Selection of "Ganache Provider" is made to a network provider.

Selection of testnet network provider: Sepolia testnet is chosen from MetaMask environment among many networks provider sets to provide a connection.

Selection of testnet network provider: Sepolia testnet is selected and matched from Remix IDE environment while Blockchain Smart Contract is written.

Bridge program Development: Writing an interaction program using IDEs and programming languages to start an interaction between Blockchain Ethereum local networks and Web3.0. For example: Truffle configuration file, etc.

Compilation and Deployment of Blockchain Smart Contract: Blockchain smart contract program is compiled and deployed to prepare smart contract ABI (JSON file format) and smart contract address, respectively. When a solidity program is deployed, there will be Blockchain smart contract program deployment. ABI is used to interact Web3.0 and local Blockchain Ethereum network. JSON file is located inside Smart Contract ABI address. It is a set of arrays of data about the compiled Blockchain Smart Contract program.

Insertion of Blockchain Smart Contract ABI file and Smart Contract address: Insertion of Blockchain smart contract ABI file and smart contract address into the web3.0 interaction program to start an interaction between Blockchain Ethereum local networks and web3.0.

Enabling Ganache CLI: Enabling Ganache CLI using npm terminal for Blockchain Ethereum local testnet network preparation.

Interaction Program Development: Development of Blockchain Ethereum local network and web3.0 interaction program is developed using Python and HTML for frontend; Solidity and CSS for backend, and JavaScript for Ganache truffle configuration. After the above tasks are done completely and properly, the Blockchain Ethereum local network and web3.0 interaction program was run using virtual (cloud-based IDE) Google CoLab IDE and Juyter Notebook IDE (local or offline IDE).

DoS Attack Protection Interactive User Dashboard Components and their Descriptions: Normal network traffic (i.e., Benign) and DoS attacks are generated one after the other when the DoS attack protection system is enabled or powered. Normal network traffic is generated immediately and continuously by default when the system is powered on. DoS attack needs an intervention (i.e., user) to be generated; otherwise, it is inactive. The interactive user dashboard acts as a client-server network environment, because it creates a suitable user interactive and real-time conditions and requirements to be simulated for the generation of both normal network traffic and DoS attacks.

User Interactive Dashboard: It is a result of the above multiple and continuous processes. All types of DoS attacks and normal network traffic are simulated and located here in the dashboard. It consists of subparts like, DoS Attack Blocked IP Addresses, Active DoS Attacks, Normal Network Traffic, CPU Utilization, Memory Utilization, Network Traffic Monitor, Recent DoS Attacks, and Recent Normal Network Traffic. All the above subsections of the dashboard display real-time numerical (i.e., name of DoS attack type and normal network traffic) data **(Figure 5)**.

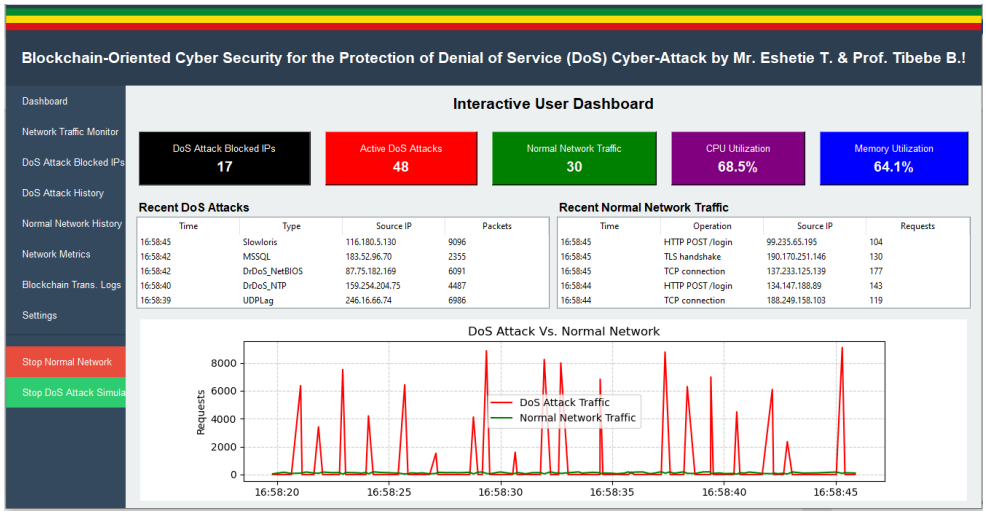


FIGURE 5: Interactive user dashboard.

Network Traffic Monitor button: The number of DoS attacks and normal network traffic is simulated, compared, and contrasted in terms of visual aid.

It presents a comparison of data analysis using bar graph between DoS attack and normal network traffic using a bar graph. A DoS attack is denoted or indicated marked using red color. Normal network traffic is indicated and marked using green color. If the number of client requests is less than 1000, normal network traffic works its normal work properly without any problem. If the number of client requests is greater than or equal to one thousand, the DoS attack red line grows from 1000 requests until it reaches 10000 requests, and then an automatic Blockchain-based DoS attack blocking mechanism is applied to decrease the attack impact. Therefore, the DoS attack red line decreases from 10000 to approximately less than 1000 number of client requests, as shown in the following figure **(Figure 5)**, and both start and end points of the green line are upper than that of the red line start and end points. This shows the ability of Blockchain to block and protect DoS attacks at that phase of the given simulation.

DoS Attack Blocked IP Addresses button: It counts and displays a set of IP addresses of DoS attacks and the date when the DoS attack occurred in this format: year-month-day at the time of the simulation phase. If the number of client requests is greater than or equal to 1000, the client IP address will be blocked in the Blockchain environment. If the number of requests is greater than or equal to the threshold, it is considered a DoS attack. Otherwise, it is considered normal network traffic.

Active DoS Attacks button: It counts and displays a numerical data about active DoS attacks which are not blocked. It includes the number of Recent DoS Attacks at the time of simulation phase.

Normal Network Traffic button: It counts and shows numerical data about normal network traffic and it includes Recent Normal Network Traffic at the time of simulation phase.

DoS Attack History button: It describes a lot about the whole generated DoS attacks, such as date (year-month-day-hour-minute-second), DoS attack type, source IP, packets and status at the time of simulation phase.

Normal Network History button: It describes a lot about the whole generated normal network, such as date (year-month-day-hour-minute-second), operation, source IP, packets, and request at the time of simulation phase.

Client Server Metrics button: It talks about performance evaluation parameters like CPU utilization and memory utilization. It represents their value in terms of a bar graph to easily understand the value of CPU and memory used during the simulation phase. CPU utilization counts and displays the amount of CPU used in percentage, and memory utilization counts and displays the amount of memory used in percentage at the time of the simulation phase.

Blockchain Logs button: It shows detailed information about Blockchain connection status, Blockchain smart contract address, and it also displays tabular information when a DoS attack is occurred and blocked. For example, date (year-month-day-hour-minute-second), DoS attacks source IP addresses, DoS attack types and packets size.

Settings button: It has two subparts: Blockchain Settings and Blockchain Notification Settings. Blockchain Settings contains detailed information about Ganache truffle or Ganache CLI (i.e http://27.0.0.1:8545) and Blockchain smart contract address (i.e 0xd9145CCE52D386f254917e481eB44e9943F39). Blockchain Notification settings have e-mail address container or holder. Suddenly, if a DoS attack is occurred, an alert or notification will be sent through the already provided email address to the concerned personnel (network administrator).

Normal Network simulation button: When this button is pressed, it will stop generating normal network traffic. It is active since the DoS attack protection system is powered on and so, no need of pressing this button.

DoS Attack simulation button: When this button is pressed, it will be enabled or activated, and it starts generating DoS attack network traffic. It is passive since even if the DoS attack protection system is powered on, there is a need of pressing this button so as to generate DoS attacks.

4.2. Discussions

The previous related works are different from this study because this study consists of eight additional parameters, such as DoS attack minimization and prevention capability, latency, throughput, CPU utilization, memory utilization, bandwidth utilization, ROC, and AUC curves with their corresponding numerical values. The previous related works focus on a few evaluation parameters, and they do not focus on other parameters. In our study, we have seen that the value of DoS attack detection accuracy and DoS attack minimization and prevention capability were 99% and 100%, respectively, using a CNN model. We use a single DoS attack dataset to train different models that produce different parameter values. For example, the RF model produced values of DoS attack detection accuracy and DoS attack minimization and prevention capability of 99% and 83.83%, respectively. At the DoS attack detection and protection process, the average latency of eight models was 48.4811 milliseconds.

We have seen that our proposed Blockchain-based Cyber Security study have scored 100% DoS attack minimization and prevention capability and 98.83% DoS attack detection accuracy using CNN model on the dataset, which was the best one result.

In the previous related existing literature, they used RF and XGBoost models, and then the value of DoS attack detection accuracy was 99.99999% and 99.99996%, respectively [15]. DoS attack detection accuracy is 96%, and AI has a total DoS attack accuracy of 80% for the identification and prevention of DoS attacks, more than the traditional method. A combination of DL and Big Data improves the detection accuracy of DoS attacks with a value of 99.78% [17].

5. Conclusions and Recommendations

This section of our study deals with the generalized and concise concepts and suggestions or feedbacks based on the DDoS attack detection and DDoS attack prevention simulations result. This part of the study, points out a hint to researchers to extend the functionality of this study by filling out the limitations and by adding additional features.

5.1. Conclusions

This study is designed to tackle DoS attacks by applying Blockchain technology, and we got attractive results as was proposed. Mixed or combined (i.e, Hybrid) models have better DoS attack detection accuracy than single models' DoS attack detection accuracy. According to our Blockchain-based proposed study, client requests could not reach directly into the client-server network layer (i.e, Server) before the fulfillment of a prerequisite condition at the Blockchain network layer. Even if the CIC-DDoS2019 dataset is available freely at the Kaggle official website, there are some drawbacks to it. Among eight models, CNN has an amazing DoS attack minimization and prevention capability of 100%, while the remaining models have below 100% of DoS attack minimization and prevention capability.

5.2. Recommendations

5.2.1. Recommendations for researchers

This study works best if it is implemented in a real-world environment to avoid a DoS attack from it. To do this, hardware and software resources are needed, and they are very expensive.

5.2.2. Recommendations for source of CIC-DDoS219 dataset

The CIC-DDoS2019 dataset has a class imbalance nature. So, the new version of balanced and light-weight or small-sized CIC-DDoS2019 dataset should be developed. The large size of this CIC-DDoS2019 dataset should be reduced to save computational resources.

Funding

This research paper is based on a theoretical concept and publicly available resources. There was no funding.

Ethical and Competing Interest

We, the authors, declare that we have no well-known competing economic interests or individual relationships that may have occurred to affect this work. So, we have nothing to declare about competing interests and any other interest, such as our employment, consultancy, stock ownerships, honorary, paid expert, patent application/registration, grants (any other funding), affiliation, etc. Therefore, there are no competing interests in this paper.

Clinical Trial Number

There was no clinical trial number. This research paper is focused on the conceptual and theoretical aspects. A clinical trial number does not apply to this paper.

Consent to Participate

All authors contributed to and participated in this research paper. For the successful completion of this paper, we fulfilled our individual responsibilities.

Consent to Publish

We, the authors, declare that we have the necessary permission to allow this research paper to be published.

References

[1] C. Dannen, *Introducing Ethereum and Solidity*, Apress Berkeley, 2017.

[2] R. Greene, M. N. Johnstone, “An Investigation into a Denial-of-Service Attack on an Ethereum Network,” *in Proceedings of the 16th Australian Information Security Management Conference*, pp. 90-96, 2018.

[3] F. Hussain, S. G. Abbas, M. Husnain, U. U. Fayyaz, F. Shahzad and G. A. Shah, “IoT DoS and DDoS Attack Detection using ResNet,” *in 2020 IEEE 23rd International Multitopic Conference*, 2020.

[4] A. Alabdulatif, N. N. Thilakarathne and M. Aashiq, “Machine Learning Enabled Novel Real-Time IoT Targeted DoS/DDoS Cyber Attack Detection System,” *Computers, Materials and Continua*, vol. 80, no. 3, pp. 3655-3683, Dec 2024.

[5] H. H. Adelsberger, B. Collis, and J. M. Pawlowski, *Handbook on information technologies for education and training*, 2nd ed, Springer Science & Business Media, 2008.

[6] L. L. Peterson and B. S. Davie, *Computer networks: a systems approach*, 4th ed, San Francisco: Elsevier, 2007.

[7] S. K. Sharma, "A Textbook for Grade 11," University of Liberia.

[8] F. CISA, "Understanding and Responding to Distributed Denial of Service Attacks," 2024.

[9] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, “DDoS attack protection in the era of cloud computing and Software-Defined Networking,” *Computer Networks*, vol. 81, pp. 308-319 Apr 2015.

[10] M. D. Pierro, “What Is the Blockchain?” *Computing in Science & Engineering*, vol. 19, no. 5, pp. 92-95, 2017.

[11] B. B. Gupta and A. Dahiya, *Distributed Denial of Service (DDoS) Attacks: Classification, Attacks, Challenges and Countermeasures*, CRC press, 2021.

[12] H. Richard, "Fundamentals of C++ Programming," 2018.

[13] A. Amjad, T. Alyas, U. Farooq, and M. A. Tariq, "Detection and mitigation of DDoS attack in cloud computing using machine learning algorithm," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 6, no. 23, 2019.

[14] J. D. Ndibwile, A. Govardhan, K. Okada, and Y. Kadobayashi, "Web server protection against application layer DDoS attacks using machine learning and traffic authentication," *in 2015 ieee 39th annual computer software and applications conference*, 2015, vol. 3, pp. 261–267.

[15] A. Golduzian, "Predict and prevent ddos attacks using machine learning and statistical algorithms," 2023.

[16] T. Islam, M. I. Jabiullah, and D. M. H. Abid, "Ddos attack preventing and detection with the artificial intelligence approach," *Intelligent Computing Systems*, vol. 1569, pp. 30–43, 2022.

[17] Y. Shang, "Prevention and detection of DDOS attack in virtual cloud computing environment using Naive Bayes algorithm of machine learning," *Measurement: Sensors*, vol. 31, p. 100991, Feb 2024.

[18] S. Yu, *Distributed denial of service attack and defense*, New York: Springer New York, 2024.

[19] S. Asri and B. Pranggono, "Impact of distributed denial-of-service attack on advanced metering infrastructure," *Wireless Personal Communications*, vol. 83, no. 3, pp. 2211–2223, 2015.

[20] FREEMAN, "The History of the Blockchain and Bitcoin," FREEMAN.

[21] D. Yaga, P. Mell, N. Roby, and K. Scarfone, "Blockchain technology overview," 2019.

[22] K. Dugan, Blockchain for Beginners: *Understand the Blockchain Basics and the Foundation of Bitcoin and Cryptocurrencies*, CRB Publishing, 2018.

[23] L. L. Svekis, M. van Putten, and C. B. R. Percival, *JavaScript from Beginner to Professional: Learn JavaScript quickly by building fun, interactive, and dynamic web apps, games, and pages*, Packt Publishing Ltd, 2021.

[24] I. Priyadarshini, "Introduction to blockchain technology," *Cyber security in parallel and distributed computing: concepts, techniques, applications and case studies*, pp. 91–107, 2019.

[25] V. Gramoli, *Blockchain Scalability and its Foundations in Distributed Systems*, Springer Cham, 2022.

[26] W. Gan, Z. Ye, S. Wan, and P. S. Yu, "Web 3.0: The future of internet," *in Companion Proceedings of the ACM Web Conference 2023*, 2023, pp. 1266–1275.

[27] A. Mansurali, K. N. Reddy, T. Chakraborty, and K. Jyothish, "Metaverse, NFTs, and Web 3.0: Applications, Challenges, and Future Opportunities," *in Understanding the Metaverse: Applications, Challenges, and the Future: Springer*, 2024, pp. 45–66.

[28] W. Ma and K. Huang, *Blockchain and Web3: Building the cryptocurrency, privacy, and security foundations of the metaverse*, John Wiley & Sons, 2022.

[29] R. M. A. Latif, K. Hussain, N. Jhanjhi, A. Nayyar, and O. Rizwan, "A remix IDE: smart contract-based framework for the healthcare sector by using Blockchain technology," *Multimedia tools and applications*, pp. 1–24, 2020.

[30] D. R. Mérette, M. Ross, T. R. Mérette, and K. Church, "Introduction to Anaconda and Python: Installation and setup," *The Quantitative Methods for Psychology,* vol. 16, no. 5, pp. S3–S11, 2016.

[31] E. Díaz García and E. Cabrera Granado, "*Guide to Jupyter Notebooks for educational purposes*," 2018.

[32] R. Modi, *Solidity Programming Essentials: A beginner's guide to build smart contracts for Ethereum and blockchain*, Packt Publishing Ltd, 2018.

[33] M. Dawson, *Python programming for the absolute beginner*, Boston: Course Technology, 2010.

[34] J. M. Ortega, *Mastering Python for Networking and Security*, 2nd ed. Packt Publishing Ltd, 2018.

[35] U. K. Prinz and P. Prinz, *A complete guide to programming in C++*, Canada: Jones & Bartlett Learning, 2002.

[36] J. McPeak, *Beginning JavaScript*, 5th ed. John Wiley & Sons, 2015.

[37] L. L. Svekis, M. V. Putten and R. Percival, *JavaScript from Beginner to Professional*, Birmingham: Packt Publishing, 2021.

[38] R. Schifreen, "The web book: how to create web sites and applications with html, css, Java script, php and mysql," 2009.

[39] R. Schifreen, "How to create Web sites and applications with HTML, CSS, Javascript, PHP and MySQL," in The Web Book, Oakworth Business Publishing Ltd, 2009.

[40] T. Powell, *HTML & CSS: the complete reference*, 5th ed. McGraw-Hill Companies, 2010.

[41] J. Dean, *Web Programming with HTML, CSS, and JavaScript*, 2th ed. Jones & Bartlett Learning, 2025.

[42] W. Haniyah, M. C. Hidayat, Z. F. I. Putra, V. A. Pertama, and A. Setiawan, "A Simulasi Serangan Denial of Service (DoS) menggunakan Hping3 melalui Kali Linux," *Journal of Internet and Software Engineering*, vol. 1, no. 2, pp. 8–8, 2024.

[43] A. Khan, P. Nand, B. Bhushan, A. A. Hameed, and A. Jamil, "A Review of Blockchain based Decentralised Authentication Solutions and their improvement through Metamask," *in 2024 2nd International Conference on Artificial Intelligence, Blockchain, and Internet of Things (AIBThings)*, 2025.

[44] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," *in Modeling and tools for network simulation*, K. Wehrle, M. Güneş, J. Gross, Springer, 2010, pp. 15-34.

[45] B. Zhang, T. Zhang, and Z. Yu, "DDoS detection and prevention based on artificial intelligence techniques," *in 2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, pp. 1276–1280.

[46] Genie Network Security Response Team, *DDoS Protection with Machine Learning and Artificial Intelligence*, Genie Networks Limited, 2019.

[47] A. Coscia, V. Dentamaro, S. Galantucci, A. Maci and G. Pirlo, "Automatic decision tree-based NIDPS ruleset generation for DoS/DDoS attacks", *Journal of Information Security and Applications*, vol. 82, May 2024.

[48] C. R. Kothari, *Research methodology: Methods and techniques*, 2th ed. India: New Age International, 2004.

[49] Kaggle, *CIC-DDoS2019 30GB (Full Dataset CSV Files)*, Kaggle.

[50] D. Lv, Z. C. Ma, S. Yang, X. Li, Z. Ma, and F. Jiang, "The application of SMOTE algorithm for unbalanced data," *in Proceedings of the 2018 International Conference on Artificial Intelligence and Virtual Reality*, 2018, pp. 10–13.

[51] D. J. Weiss, K. Brennan, R. Thomas, A. Kirlik, and S. M. Miller, "Criteria for performance evaluation," *Judgment and Decision Making*, vol. 4, no. 2, pp. 164–174, 2009.

[52] J. I. Myung and M. A. Pitt, "Model comparison methods," *Methods in Enzymology*, vol. 383, pp. 351–366, 2004.

[53] I. T. Hawryszkiewycz, *Introduction to systems analysis and design*, Prentice-Hall, Inc., 1988.

[54] A. Dennis, B. H. Wixom and R. M. Roth, Systems analysis and design, 5th ed. John wiley & sons, 2008.

[55] Quora. "What is the typical range (number) of requests to a web server that can cause distributed denial of service attacks?"