IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 3, June, 2024, pp. 1043-1049
Journal homepage: http://iieta.org/journals/isi

Detection and Mitigation Distribution Denial of Service Attack Based on Blockchain Concept

Alharith A. Abdullah*, Shahad A. Hussein

College of Information Technology, University of Babylon, Babil 51002, Iraq

Corresponding Author Email: alharith@uobabylon.edu.iq

https://doi.org/10.18280/isi.290322

Received: 5 October 2023
Revised: 8 May 2024
Accepted: 30 May 2024
Available online: 20 June 2024

Keywords:
Distributed Denial of Service (DDoS) attacks, TCP-based attacks, attack detection, blockchain technology, smart contracts

ABSTRACT

Distributed Denial of Service (DDoS) attacks are security threats that attempt to disrupt network service provisioning by overwhelming it with a massive flood of data traffic, with the goal of exhausting network resources like bandwidth. Attackers can easily employ a lot of devices as part of botnets to launch DDoS attacks. Thus, DDoS attacks prevent users from accessing required services, and it is also possible for them to crash the entire network. The solution proposed in this paper utilizes blockchain technology and reversed connection requests to prevent TCP attacks, which are a common form of DDoS attack. This mechanism aims to protect the network from crashing and ensure uninterrupted service provision. The experimental evaluation achieved success with our system, where the attack detection rate was above 99%. Additionally, the accuracy percentage of detecting the attack was also above 99%. Moreover, we succeeded in achieving low false positive rates and false alarm rates with the same low percentage. Finally, the process of discovering the attacking address took about 0.749 seconds.

1. INTRODUCTION

A distributed denial-of-service attack is defined as a form of attack that destroys services that are provided over the Internet, through exploiting the inherent vulnerabilities of the Internet's infrastructure [1]. Where these cyberattacks in the current interconnected world have become an important problem that causes concern among many groups, such as individuals, companies, and governments [2]. Accordingly, the primary aim of a DDoS attack is to disrupt and confuse online services by overloading and weakening them, resulting in the prevention of authorized users from accessing the target that offers these services [3]. Where, the attacker launches a synchronized attack that overwhelms the target's bandwidth, processing power, or application layer resources [4] and thus, the distributed denial-of-service attack depends in its essence mainly on exploiting a network of compromised devices, such as computers, smartphones, and cameras, known as botnets, in order to flood the target network or servers with a huge amount of data traffic [5]. This large flood of data traffic consumes the resources available in the target, preventing the systems from dealing with and responding to authorized requests [6].

DDoS attacks come in a variety of forms, such as volumetric attacks that overwhelm the target's network with a large volume of traffic, application layer attacks that target a particular application or service, and TCP-based attacks that exhaust the TCP connection's stateful nature to deplete server resources [4]. TCP-based attacks are one of the most common types of DDoS attacks [7]. These attacks create a huge number of TCP connections or half-open ones, where the attacker sends a massive number of TCP SYN packets (synchronization packets) to the target system for the purpose of simulating the initial legitimate connection phase of the three-way handshake process [8]. However, this process is not fully completed as the attacker does not send the decision packet (acknowledgment), which is the last stage of the connection [9]. This causes the target system to flood with half-open connections that consume system resources and makes it unable to process legitimate requests [10]. Moreover, DDoS attacks can also be divided into groups according to where they come from, such as network-layer attacks, where traffic comes from numerous IP addresses, or application-layer attacks, which take use of flaws in particular programs [6, 11].

On the other hand, there are many ways through which these attacks can be detected, prevented, or mitigated [12]. One such way is network-based detection techniques whereby network traffic is monitored, and abnormal patterns of this traffic are identified. Another one is traffic analysis which is considered one of the network-based approaches [11, 13, 14]. Additionally, machine learning and artificial intelligence techniques are widely used today to detect distributed denial-of-service attacks, in addition to blockchain technology [2, 4, 15, 16]. Intelligence techniques are widely used today to detect distributed denial-of-service attacks, In addition to blockchain technology [2, 4].

In cybersecurity, blockchain technology is becoming a trend nowadays among favorable technologies [2]. It enables storing records in a decentralized and transparent manner, as it is a peer-to-peer network at its core that consists of a distributed ledger called blocks [17]. These blocks use cryptography to link to each other, and each block contains the information of its previously linked block, such as cryptographic hash, unique identifier, nonce, Merkle tree, and timestamp [17]. This creates an information chain that is immutable and tamper-proof [5, 11, 18]. Accordingly, Blockchain provides advantages through the usage of smart contracts, including automation, decentralization, transparency, and security, making it difficult for unauthorized parties to manipulate or alter data [19-21].

This research proposes a DDoS detection service (DDoS_DS) based on statistical methods for real-time network traffic analysis. The detection method relies on smart contracts that initiate reverse connection requests using key features for network scanning capabilities. This enables the detection of open ports and device identification, facilitating the detection of DDoS attacks in a network environment. Additionally, it aims to mitigate the impact of these attacks by using one of a protection strategy.

The use of a smart contracts on blockchain technology to detect a distributed denial-of-service (DDoS) attack has successfully achieved attack detection, as well as accuracy detection, at a rate above 99%, with low rates for false positives and false alarms. Moreover, the time for the detection process was approximately 0.749 seconds. These contracts were able to extract the sender's IP address and analyze whether it was a normal address or an attacker's address, and then create a blocklist of spoofed IP addresses. Finally, this method made it possible to mitigate the impact of DDoS attacks.

The upcoming section will discuss the research conducted prior to this work. The proposed method for detecting and preventing DDoS attacks will be presented in Section 3. The fourth section will provide a summary of the findings and conclusions of this work.

2. RELATED WORK

Various techniques and approaches are employed to detect and mitigate DDoS attacks in diverse networks. These include statistical methods, machine learning, Blockchain-Based Solutions, and other security technologies. The most suitable method to use depends on the network structure and the potential types of attacks.

Researchers have proposed various methods to address DDoS attacks on networks. In the study by Kumar and Gowda [6], the research addresses the issue of DDoS attacks on IoT systems by proposing a model that utilizes the Ethereum blockchain. The proposed model not only tackles the problem of single points of failure, privacy, and security in IoT systems but also provides a decentralized platform for preventing DDoS attacks at the application layer. The authentication and verification of IoT devices are carried out to prevent malicious devices from connecting and communicating with the IoT networks. The IP addresses of these malicious devices are traced and recorded inside the blockchain. The system's performance was evaluated through 100 experiments, which demonstrated its superiority over other related works due to fewer I/O operations, resulting in faster execution time.

Meanwhile, an architecture and design for a collaborative mechanism was introduced by Rodrigues et al. [1]. This mechanism takes advantage of using smart contracts to counter DDoS attacks in a fully decentralized manner. The proposed approach distributes rules to signal white or blacklisted IP addresses across multiple domains and utilizes the features of smart contracts to filter traffic. Even if the victim AS (Autonomous System) does not implement these rules, the method can still effectively mitigate the attack.

A modern method to defend against DDoS attacks on IoT devices has been proposed in a recent paper [22], which is based on blockchain technology. This method involves extracting features of the network traffic from edge nodes, analyzing the data, and identifying abnormal behavior on terminal devices by deploying smart contracts within the blockchain network to defend against attacks. This method is able to quickly identify potential attacks through early detection of abnormal behavior. Additionally, the use of smart contracts ensures that DDoS attack node information and access control strategies are synchronized, preventing network congestion and providing an advantage in defending against attacks.

On the other hand, Manikumar et al. [4] proposed a system that utilizes machine learning algorithms to distinguish incoming packets as malicious or not and creates a blacklist using blockchain technology. Effectively storing blacklisted IP addresses makes the blockchain provide an additional layer of security over existing DDoS mitigation systems. The Tree-Based Classifier algorithm was used for feature selection to enhance computational time. Real-time traffic analysis showed that the Random Forest algorithm provided an accuracy of approximately 95%.

3. THE PROPOSED METHODOLOGY

The proposed mechanism uses the principles of blockchain technology to detect DDoS attacks and protect the network from their impact. One of these principles that has been used is decentralization, as the nodes in the network are connected to each other according to the peer-to-peer architecture. Therefore, this principle frees the system from a single point of failure. In addition, the principle of a distributed ledger in this network is used to achieve sharing and synchronization of the operations that occur within the nodes, and the last principle is smart contracts that have been relied upon in the network. Figure 1 shows the workflow of the proposed system.

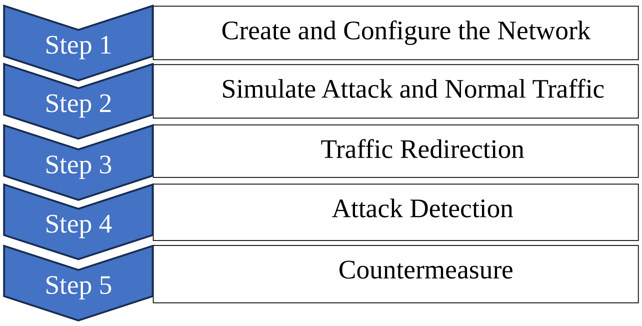


Figure 1. The proposed system flow work

Step 1: Create and Configure the Network: The proposed system's Network component consists of two types of nodes:

(1) The main node, known as the Server, and the auxiliary nodes, represented by smart contracts deployed on a blockchain network. The Server receives and responds to client requests and employs measures to maintain network performance when its processing capacity is exceeded.

(2) The auxiliary nodes are directly connected to the Server

and come into play when the Server's processing capacity is overwhelmed. These smart contracts possess specialized systems to detect the sender of a request, distinguishing between normal clients and potential attackers. They do not directly respond to client requests but instead take action against the sender, either by blocking or allowing the requests.

The network comprises nine auxiliary nodes, but only four of them are connected to the Server. Additionally, all nodes are interconnected using the mentioned architecture, as depicted in Figure 2.

Assuming the server's processing capacity is 1000 requests per second on average, there are three normal IP addresses and three spoofed IP addresses. The system's behavior is illustrated through a flowchart in Figure 3.

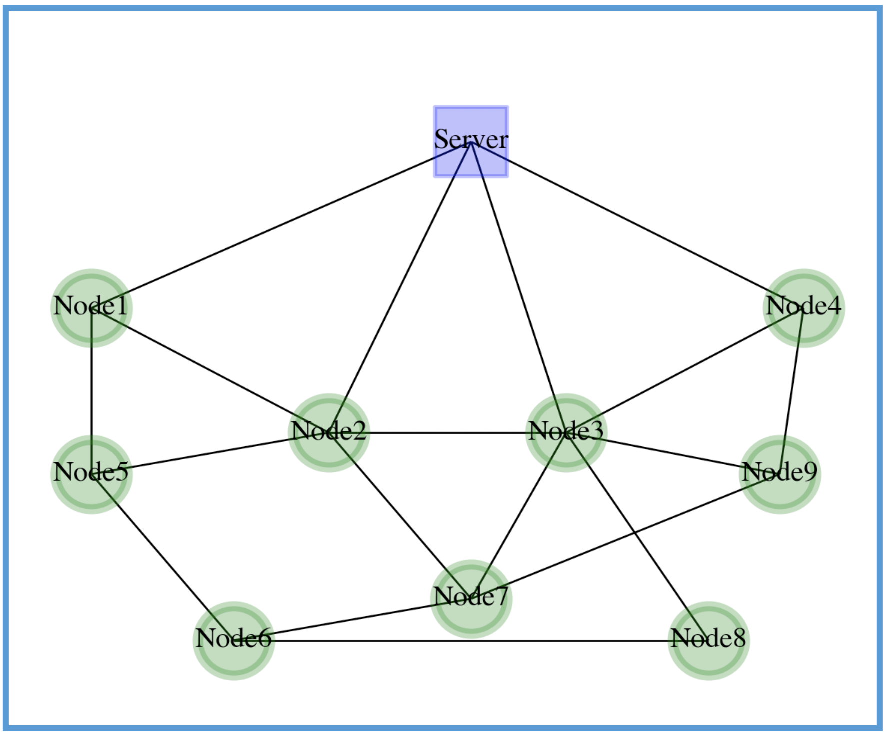


Figure 2. Peet-to-Peer Node connection of the proposed system

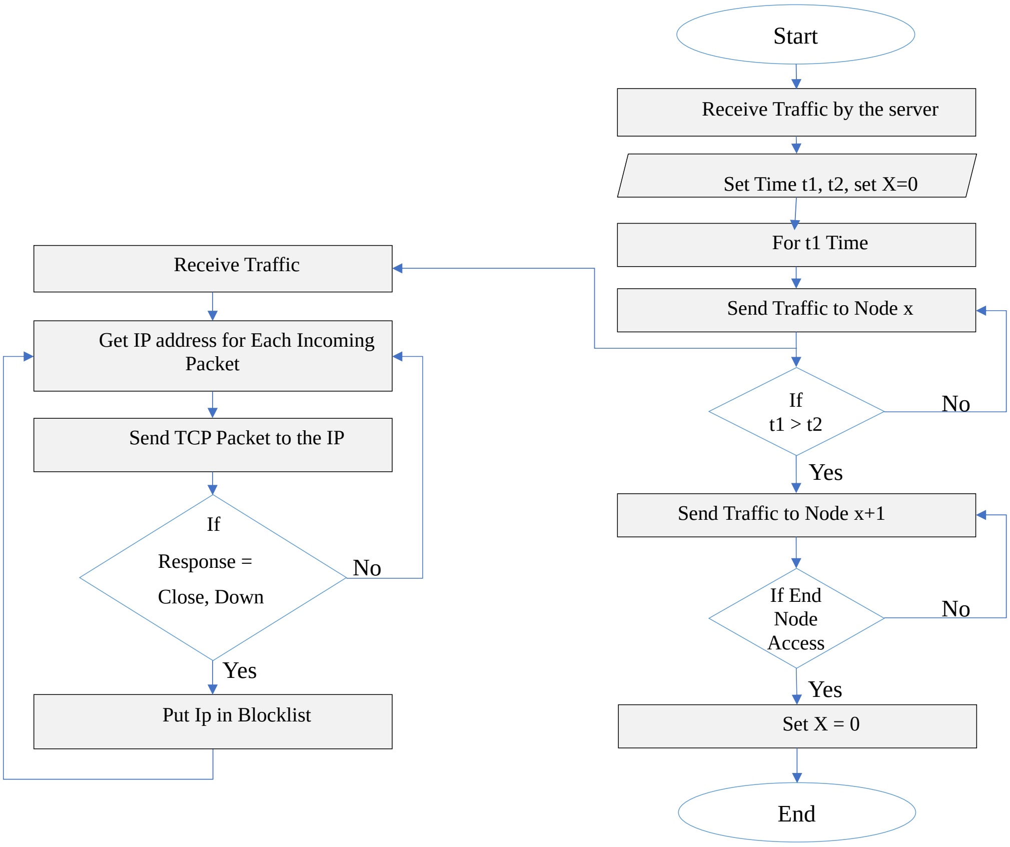


Figure 3. Flowchart of the proposed system behavior

Step 2: Simulate Attack and Normal Traffic : The attack and normal traffic are simulated programmatically, by generating traffic with attack features. The attack focuses on a single target, meaning that the receiver address in all packets is the same. Moreover, the inter-arrival time of the attack traffic is very short, in addition to the non-response characteristic, as the attack only sends a request packet (SYN packet). While normal traffic does not have any of the mentioned attack properties.

In this step of the proposed work, traffic was generated a mixture of normal traffic and the attack, then was sent to the server node.

Step 3 and Step 4: Traffic Redirection and The Attack Detection: The process of redirecting requests is carried out by distributing them in a round-robin manner among the nodes for a certain period. The server's actions involve redirecting packets to adjacent nodes to detect for any signs of a DDoS attack. Each adjacent node receives a set number of packets every 5 seconds in a round-robin fashion. During each iteration, when a node receives a packet, it performs a detection process. It extracts the source IP of each packet and sends a TCP Probe packet to that IP address. If the node receives a response indicating that the output is open or up, it considers the IP address to be normal. If there is no response indicating that the output is closed or down, the node identifies the IP address as a spoof IP. Figure 4 provides an explanation of the detection procedure.

Step 5: The Countermeasure: After the nodes discover the attacking addresses, they add these addresses to the blacklist. Additionally, any of these nodes distribute this list to the rest of the nodes, and inform the server of this list immediately after discovering each new address.

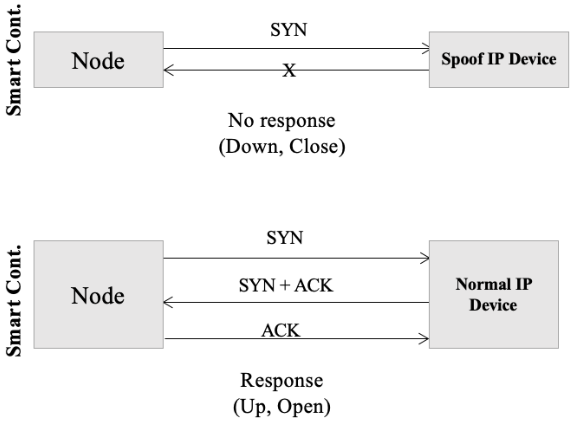


Figure 4. Discovering process of normal and attack traffic

4. RESULTS AND DISCUSSION

The proposed system was built using by using python programming language v3.9, PyCharm community 2021.3 environment, and a Mac operating system with the processor air M1, and memory of 8G in the following manner:

4.1 Case 1: In the absence of smart contracts

The server receives several requests that match its capability and process them. Similarly, it will respond to requests normally when the requests do not exceed its capability. However, if the server receives requests exceeding its capability to handle them the server will start to fail whenever it exceeds the server's capacity limit and then stop working completely. The implementation results of the proposed system in this scenario are presented in Table 1.

Table 1. The implementation statistics of the proposed system in case of absence the smart contracts

Requests Rate	Failer Rate
50	1%
100	3%
1000	8%
2000	12%
24000	100%

From the table above we notice that the failure rates are 1%, 3%, and 8%, these percentages indicate the state of the server in processing requests and are usually due to the state of the network in addition to the time required to process each request, which in turn depends on the type of service requested.

If the normal rate of the server's ability to process requests is exceeded, we notice that the failure rate begins to increase to 12 percent, and as the percentage of requests increases significantly, this leads to the server reaching a state of complete failure (100%) due to the server's processing capability being exceeded.

4.2 Case 2: In the existence of smart contracts

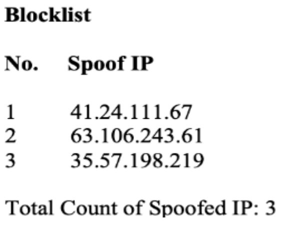
When the server receives more requests than its capacity, it will periodically redirect the requests to auxiliary nodes connected to it for five seconds in each cycle. Each node will then examine the request's source. If a spoofed IP is detected, it will be blocked, reducing the number of requests on the server. Table 2 displays the implementation results of the proposed system in this scenario.

From the table above, we note that the failure rates are 1% and 3% remaine, for the same reason above, since in this case the smart contract was not used due to the server's ability to process requests not being exceeded.

While the failure rate decreased to less than 3 percent when the rate of requests was greater than or equal to 1000, that is, in the case of exceeding the server's ability to process requests. The reason for this decrease is due to the presence of the smart contract, as the server, in this case, forwarded the requests to the smart contract to check its state, thus able to detect and prevent the attack by using the blocklist.

Table 2. The implementation statistics of the proposed system in case of existence the smart contracts

Requests Rate	Failer Rate
50	1%
100	3%
Greater than or equal to 1000	Less than 3%


Blocklist

No.	Spoof IP
1	41.24.111.67
2	63.106.243.61
3	35.57.198.219

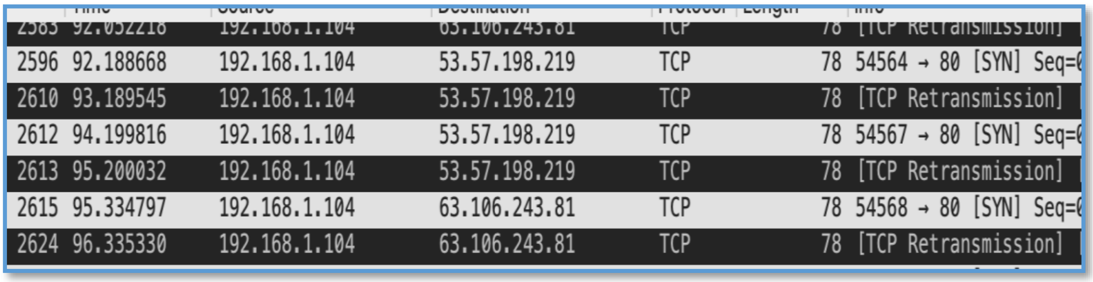
Total Count of Spoofed IP: 3

Figure 5. The blocklist created by smart contract

Thus, the blocklist now contains a set of IP addresses for suspicious requests that were identified by the smart contract, and this list is distributed at every update to all smart contracts in addition to the server. Figure 5 shows the blocklist that contains the spoofed Source IP addresses that were detected by the smart contract.

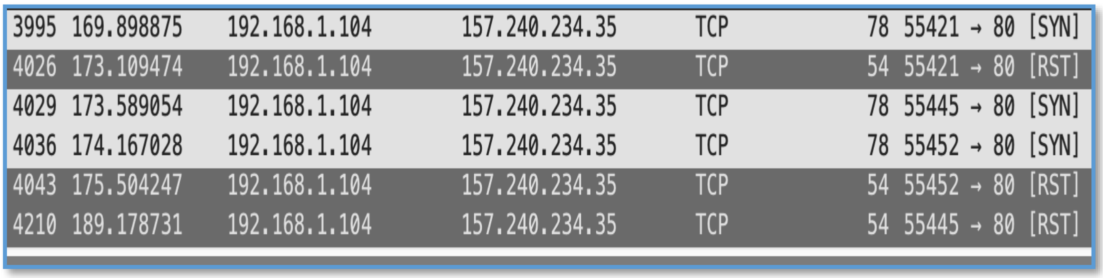
In addition, the results were clarified using the Wireshark application. It is clear from the application output that spoofed IP addresses were used for the attack. It is noted that after sending a request using TCP Probe (SYN) to these IP addresses, the output remains as (TCP Retransmission) until timeout, indicating that there is no response to this request. As shown in Figure 6.

While analyzing the Wireshark output, it was observed that when the node sends a TSP prop (SYN) to a normal address, the response received is (SYN+ACK). Subsequently, the node sends a (RST) packet, which is the third step in the communication process of the three-way handshake. This message indicates the completion of the connection between the two ends and hence, confirms that the address is a normal one and not an attacker's address. Refer to Figure 7 for a visual representation.



```
     Time       Source          Destination     Protocol Length  Info
2583 92.052218  192.168.1.104   63.106.243.81   TCP          78  [TCP Retransmission]
2596 92.188668  192.168.1.104   53.57.198.219   TCP          78  54564 → 80 [SYN] Seq=
2610 93.189545  192.168.1.104   53.57.198.219   TCP          78  [TCP Retransmission]
2612 94.199816  192.168.1.104   53.57.198.219   TCP          78  54567 → 80 [SYN] Seq=
2613 95.200032  192.168.1.104   53.57.198.219   TCP          78  [TCP Retransmission]
2615 95.334797  192.168.1.104   63.106.243.81   TCP          78  54568 → 80 [SYN] Seq=
2624 96.335330  192.168.1.104   63.106.243.81   TCP          78  [TCP Retransmission]
```

Figure 6. The results of the wireshark when sending a TSP probe to the spoofed IP address



```
3995 169.898875  192.168.1.104   157.240.234.35   TCP   78  55421 → 80 [SYN]
4026 173.109474  192.168.1.104   157.240.234.35   TCP   54  55421 → 80 [RST]
4029 173.589054  192.168.1.104   157.240.234.35   TCP   78  55445 → 80 [SYN]
4036 174.167028  192.168.1.104   157.240.234.35   TCP   78  55452 → 80 [SYN]
4043 175.504247  192.168.1.104   157.240.234.35   TCP   54  55452 → 80 [RST]
4210 189.178731  192.168.1.104   157.240.234.35   TCP   54  55445 → 80 [RST]
```

Figure 7. The results of the wireshark when sending a TSP probe to the normal IP address

5. THE EVALUATION

Based on the results obtained in the previous section, the proposed system can be evaluated according to the following criteria:

5.1 Attack detection rate

The detection rate of attacks is the ratio of detected attack packets to the total attack packets. In the proposed system's environment, a high attack detection rate of approximately more than 99% was achieved, as all packets sent by the attacker were detected.

5.2 False alarm rate and false positive rate

In the proposed system, the percentage of false alarms was very low, accessing less than 1%. This is because the method used depends on the process of establishing a connection using the three-way handshake. The same ratio applies for false positive rate due to the successful classification of normal packets in the proposed system. Therefore, the success of this connection establishment depends on the communication environment. If the environment allows for communication between the two parties, ideal results can be obtained. However, these percentages may increase if there are obstacles to establishing communication between the two parties, such as natural environmental obstacles that prevent communication.

5.3 Accuracy

The ratio between the number of packets correctly classified, whether these packets are normal or attacks, and the total number of packets. In this work, this percentage was greater than 99%, as each type of packet was classified correctly as intended. As mentioned previously, this also depends on the three-way handshake process used to determine the type of packet, which in turn depends on the communication conditions between the two parties. Therefore, the stronger and problem-free the connectivity, the higher this percentage.

5.4 Time

The time it takes to detect the packet type is considerable. While the proposed system shows high potential in detecting package types, the process is somewhat slow and takes time. Specifically, in the proposed system, it takes (0.749) seconds to discover an attacking address and (0.852) seconds to discover a normal address. This delay may cause a delay in responding to regular customer requests.

6. COMPARISON

Finally, this section conducts a comparative analysis between our proposed model for detecting Distributed Denial of Service (DDoS) attacks, specifically TCP attacks, and a selection of previous works with similar detection objectives. This comparison focused on the detection environment, methods used, and key performance metrics, such as accuracy and false positive detection rate.

Table 3 shows that the false positive detection rate in previous works based on machine learning and deep learning methods is higher than the rate achieved in our proposed model. The reason is that machine learning and deep learning methods depend on the features of the used data sets that the model trained on, and therefore the detection of attacks depends on the model resulting from the training, thus If there is traffic that the model was not previously trained on, it will give an incorrect decision.

Compared to our work, the false positive detection rate was lower due to it is interactive nature with the attacker's traffic and thus it has ability to detect the different behaiviour taken by the attacker, and this would also make the detection accuracy very high.

Table 3. Our work's evaluation in comparison to other similar works

References	Method	Environment	Result	
			FPR	Accuracy
[23]	Machine Learning-random forest algorithm -SVM method	Utilize a tool to create traffic for the purpose of training and testing the model.	0.16%	99.49%
[24]	Deep Learning based on Long Short-Term Memory (LSTM)	Data set-Cicids2019	-	99.19%
[25]	continuous ranked probability score (CRPS) statistical metric and exponentially smoothing (ES) scheme	Data Set-DARPA99, MAWI and ICMPv6	0.18%	99.91%
Our Model	Key Capabilities in Network Scanning with Smart Contracts on blockchain technology.	Network	less than 0.1%.	More than 99%

7. CONCLUSIONS

This paper proposes a mechanism to detect DDoS attacks in a network using blockchain technology. The system aims to detect flooding attacks and mitigate their effects. The system was implemented using the Python programming language. The proposed system achieved excellent results in terms of detection accuracy and false positive rate, with 0.1% false alarms in the simulated network and a very high accuracy of attack detection rate. Thus, the proposed mechanism achieves the detection of DDoS attacks as well as the protection of the network.

REFERENCES

[1] Rodrigues, B., Bocek, T., Lareida, A., Hausheer, D., Rafati, S., Stiller, B. (2017). A blockchain-based architecture for collaborative DDoS mitigation with smart contracts. In Security of Networks and Services in an All-Connected World: 11th IFIP WG 6.6 International Conference on Autonomous Infrastructure, Management, and Security, AIMS 2017, Zurich, Switzerland, pp. 16-29. https://doi.org/10.1007/978-3-319-60774-0_2

[2] Patel, V., Khatiwala, F., Choksi, Y. (2021). An approach to detect and prevent distributed denial of service attacks using blockchain technology in cloud environment. In Advances in VLSI and Embedded Systems: Select Proceedings of AVES 2019, Singapore, pp. 247-258. https://doi.org/10.1007/978-981-15-6229-7_20

[3] Khalaf, B.A., Mostafa, S.A., Mustapha, A., Mohammed, M.A., Mahmoud, M.A., Al-Rimy, B.A.S., Razak, S.A., Elhoseny, M., Marks, A. (2021). An adaptive protection of flooding attacks model for complex network environments. Security and Communication Networks, 2021(1): 5542919. https://doi.org/10.1155/2021/5542919

[4] Manikumar, D.V.V.S., Maheswari, B.U. (2020). Blockchain based DDoS mitigation using machine learning techniques. In 2020 Second international conference on inventive research in computing applications (ICIRCA), Coimbatore, India, pp. 794-800. https://doi.org/10.1109/ICIRCA48905.2020.9183092

[5] Ibrahim, R.F., Abu Al-Haija, Q., Ahmad, A. (2022). DDoS attack prevention for internet of thing devices using ethereum blockchain technology. Sensors, 22(18): 6806. https://doi.org/10.3390/s22186806

[6] Kumar, B.S., Gowda, K.K. (2022). Detection and Prevention of TCP SYN flooding attack in WSN using protocol dependent detection and classification system. In 2022 IEEE International Conference on Data Science and Information System (ICDSIS), Hassan, India, pp. 1-6. https://doi.org/10.1109/ICDSIS55133.2022.9915949

[7] Ramkumar, B.N., Subbulakshmi, T. (2021). TCP SYN flood attack detection and prevention system using adaptive thresholding method. In ITM Web of Conferences, 37: 01016. https://doi.org/10.1051/itmconf/20213701016

[8] Banu, R., Jyothi, T., Amulya, M., Anju, K.N., Raju, A., Kashyap, S.N. (2019). Monosek–A network packet processing system for analysis & detection of tcp xmas attack using pattern analysis. In 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, pp. 952-956. https://doi.org/10.1109/ICCS45141.2019.9065325

[9] Hartpence, B., Kwasinski, A. (2020). Combating TCP port scan attacks using sequential neural networks. In 2020 International Conference on Computing, Networking and Communications (ICNC), Big Island, HI, USA, pp. 256-260. https://doi.org/10.1109/ICNC47757.2020.9049730

[10] Wani, S., Imthiyas, M., Almohamedh, H., Alhamed, K. M., Almotairi, S., Gulzar, Y. (2021). Distributed denial of service (DDoS) mitigation using blockchain-A

comprehensive insight. Symmetry, 13(2): 227. https://doi.org/10.3390/sym13020227

[11] Mahjabin, T., Xiao, Y., Sun, G., Jiang, W. (2017). A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks, 13(12): 1550147717741463. https://doi.org/10.1177/1550147717741463

[12] Mohammad, H.M., Abdullah, A.A. (2023). DDoS attack mitigation using entropy in SDN-IoT environment. In AIP Conference Proceedings, 2591(1): 020002. https://doi.org/10.1063/5.0123465.

[13] Kareem, M.I., Jasim, M.N. (2022). Entropy-based distributed denial of service attack detection in software-defined networking. Indonesian Journal of Electrical Engineering and Computer Science, 27(3): 1542-1549. https://doi.org/10.11591/ijeecs.v27.i3.pp1542-1549

[14] Kareem, M.I., Jasim, M.N. (2022). Machine learning-based DDoS attack detection in software-defined networking. In International Conference on New Trends in Information and Communications Technology Applications, pp. 264-281. https://doi.org/10.1007/978-3-031-35442-7_14

[15] Kareem, M.I., Jasim, M. (2022). DDOS attack detection using lightweight partial decision tree algorithm. In 2022 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, pp. 362-367. https://doi.org/10.1109/CSASE51777.2022.9759824

[16] Oleiwi, W.K., Abdullah, A.A. (2021). Design and implementation of distributed controller clustering for solving the issue of single failure in SDN networks. Webology, 18(2): 1365-1378. https://doi.org/10.14704/WEB/V18I2/WEB18395

[17] Gamage, H.T.M., Weerasinghe, H.D., Dias, N.G.J. (2020). A survey on blockchain technology concepts, applications, and issues. SN Computer Science, 1: 1-15. https://doi.org/10.1007/s42979-020-00123-0

[18] Ahmed, Z., Danish, S.M., Qureshi, H.K., Lestas, M. (2019). Protecting iots from mirai botnet attacks using blockchains. In 2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), Limassol, Cyprus, pp. 1-6. https://doi.org/10.1109/CAMAD.2019.8858484

[19] Kim, K., You, Y., Park, M., Lee, K. (2018). Ddos mitigation: Decentralized cdn using private blockchain. In 2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN), Prague, Czech Republic, pp. 693-696. https://doi.org/10.1109/ICUFN.2018.8436643

[20] Jamader, A.R., Das, P., Acharya, B.R. (2019). BcIoT: blockchain based DDoS prevention architecture for IoT. In 2019 International Conference on Intelligent Computing and Control Systems (ICCS), Madurai, India, pp. 377-382. https://doi.org/10.1109/ICCS45141.2019.9065692

[21] Chen, M., Tang, X., Cheng, J., Xiong, N., Li, J., Fan, D. (2020). A DDoS attack defense method based on blockchain for IoTs devices. In Artificial Intelligence and Security: 6th International Conference, ICAIS 2020, Hohhot, China, pp. 685-694. https://doi.org/10.1007/978-981-15-8086-4_64

[22] Jawdhari, H.A., Abdullah, A.A. (2021). A novel blockchain architecture based on network functions virtualization (NFV) with auto smart contracts. Periodicals of Engineering and Natural Sciences, 9(4): 834-844. https://doi.org/10.21533/pen.v9i4.2441

[23] Pei, J., Chen, Y., Ji, W. (2019). A DDoS attack detection method based on machine learning. In Journal of Physics: Conference Series, 1237(3): 032040. https://doi.org/10.1088/1742-6596/1237/3/032040

[24] Shurman, M., Khrais, R., Yateem, A. (2020). DoS and DDoS attack detection using deep learning and IDS. The International Arab Journal of Information Technology, 17(4A): 655-661. https://doi.org/10.34028/iajit/17/4A/10

[25] Bouyeddou, B., Kadri, B., Harrou, F., Sun, Y. (2020). DDOS-attacks detection using an efficient measurement-based statistical mechanism. Engineering Science and Technology, an International Journal, 23(4): 870-878. https://doi.org/10.1016/j.jestch.2020.05.002