International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Flooding based Distributed Denial of Service Attacks Prevention using Blockchain Technology

Himanshu Shukla*[1], Dr Ajay Pratap[2], Dr Harsh Dev[3]

Submitted: 27/01/2024 **Revised**: 05/03/2024 **Accepted**: 13/03/2024

Abstract: The threats of Distributed Denial of Service (DDoS) attacks are a significant concern for online services. In order to prevent genuine users from accessing the target servers or networks, these assaults entail flooding them with excessive amounts of traffic. Distributed Denial of Service (DDoS) attacks pose a significant threat to the availability and reliability of online services. Among the various types of DDoS attacks, flooding-based attacks, which overwhelm a target system with a high volume of traffic, are particularly challenging to mitigate. Blockchain technology has become a more viable option in recent years for enhancing the security and resilience of network infrastructures. This paper explores the potential of Blockchain technology in preventing flooding-based DDoS attacks. By leveraging the decentralized and immutable nature of Blockchain, along with its ability to enforce consensus and facilitate secure transactions, novel approaches for detecting and mitigating flooding-based DDoS attacks can be developed. This paper provides an overview of existing techniques for DDoS attack prevention using Blockchain technology, discusses their strengths and limitations, and proposes future research directions in this area.

Keywords: *Flooding, DDoS Attacks, Blockchain Technology, Prevention, Markov Process*

1. Introduction

Denial of Service (DoS) attacks target system vulnerabilities by deliberately overwhelming them. For instance, attackers might flood the network with an excessive volume of traffic, surpassing the network card's handling capacity. Alternatively, they might inundate an application with an abundance of requests, surpassing its processing capabilities. It stems either from the network layer or the application layer within the attacker's system, both integral parts of a network. The consequences of such an attack vary, ranging from mere inconvenience in accessing specific services to severe failures within the targeted system. During periods of substantial traffic directed at a server, it becomes crucial to differentiate between genuine access and potential attacks. Here we are classifying DDoS attacks on various parameters such as use of protocol, network layer and attack mechanism.

DDoS attacks categorize broadly into three groups on the basis of protocol used which are: UDP, TCP, and miscellaneous types. These classifications hinge on the primary internet protocols leveraged for data transfer, which malicious users exploit to orchestrate attacks. The OSI/ISO model comprises seven layers, each protocol within the network associated with a specific layer. DDoS attacks target the Data Link Layer, Network Layer, Transport Layer, and Application Layer where data transmission occurs in various forms.

There are three distinct groups categorized by the attack mechanisms they employ: The first involves flooding attacks, which overload communication channels with excessive traffic. The second exploits vulnerabilities within the network protocol stack. The third group focuses on application-level attacks, targeting vulnerabilities within specific applications or services.

1.1. DDoS Attacks

DDoS assaults overload a target network or server with traffic. They're scattered, which means they come from various places. The intention is to interfere with regular operation so that authorized users cannot access it. Attackers accomplish this by employing a variety of strategies, such as SYN or UDP flooding. DDoS attack detection and mitigation call for certain instruments and methods.

1.2. Blockchain in Distributed Environment

A Blockchain serves as a distributed ledger, comprised of transactional data exchanged among multiple parties. These blocks are interlinked through secure cryptographic hashes. Each block contains the cryptographic hash of the prior block, along with a timestamp and transactional details, often presented as a Merkle Tree. The design of Blockchain inherently resists data modification, serving as an open and decentralized ledger efficiently recording transactions among entities. This network operates on a peer-to-peer basis, adhering to communication protocols

[1] *AIIT, Amity University Uttar Pradesh, Lucknow – 226012, India*
ORCID ID : 0000-0003-1328-7453
[2] *AIIT, Amity University Uttar Pradesh, Lucknow – 226012, India*
ORCID ID : 0000-0001-8124-2471
[3] *Babu Banasri Das University, Uttar Pradesh, Lucknow, India*
ORCID ID : 0000-0003-0008-6329
** Corresponding Author Email: himanshushukla@csjmu.ac.in*

for inter-node interactions and block validation. Once data is recorded within a block, it remains unalterable retroactively. Blockchain's evolution has unfolded across distinct stages or generations, marking significant modifications and advancements. Currently, it has progressed through three major generations.

The initial phase of blockchain networks, represented by Bitcoin and digital currencies, established the concept of a shared public ledger, primarily designed to support a secure digital currency within a network. After this, the second generation saw the integration of Smart Contracts, a significant addition to blockchain technology attributed to Vitalik Buterin's creation and launch of Ethereum. This advancement introduced computational abilities to the blockchain, enabling benefits in managing assets and facilitating trust agreements. Moving into the third generation, technology continuously evolves each day. In this phase, the focus lies on the intercommunication of multiple chains, forming a versatile and expansive platform that extends beyond a single blockchain structure. This evolution represents the emergence of a robust technological platform leveraging interconnected chains for diverse applications.

1.3. Limitations of Blockchain Technology

Despite its reputation for robustness and security, Blockchain is not immune to attacks. Over time, numerous attacks targeting blockchain networks as a whole have inflicted hardships on users. Various types of attacks on blockchain networks include:

- Majority attacks occur when a malicious user seizes control of over 51% of the network's block rate, constructing an alternate chain of blocks that eventually replaces the genuine one. Though often considered improbable, several cryptocurrencies based on blockchain have fallen victim to this vulnerability.

- In mining pool attacks, multiple miners combine their computational power within a mining pool. A malicious user targets this pool, aiming to compromise control, both externally and internally, by exploiting vulnerabilities in the consensus mechanism of the blockchain.

- DDoS attacks stand as the most prevalent form of assault on blockchain networks. These attacks are aimed at overwhelming a server by inundating it with an excessive volume of requests, thereby exhausting its processing resources. Attackers executing DDoS aim to disrupt the operations of mining pools, e-wallets, and other financial services linked to the network nodes. Through Distributed Denial of Service attacks, the entire blockchain network can be disrupted by inundating full node operators with a surge of traffic.

- Race attacks occur when a malicious user generates two contradictory transactions. Initially, one transaction is submitted to a service provider, which approves the transaction and initiates service delivery without confirming it. Concurrently, an uncertain transaction is disseminated to every network node, returning an equivalent amount of cryptocurrency to the malicious user. This action renders the original transaction invalid.

2. LITERATURE SURVEY

This section delves into the operational mechanisms of different Blockchain-based strategies aimed at countering DDoS attacks within cloud environments. A key aspect of our research involves a thorough analysis of these approaches, pinpointing any inherent challenges they may pose. Most of the blockchain solutions are based on either distributed architecture or access management or traffic control or Ethereum tools-based solutions as shown in Figure 3.

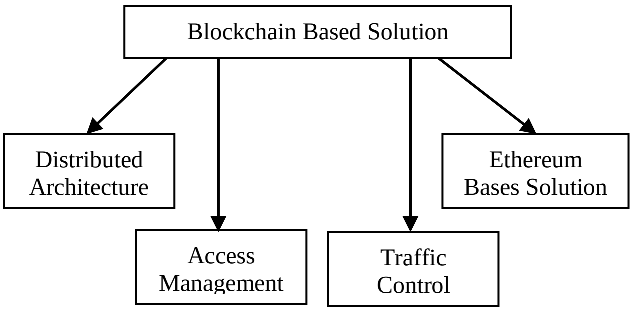

Fig. 1. Possible Solutions to Mitigate DDoS Attacks

2.1. Ethereum- based Solution

Ethereum-based solutions leverage the inherent mechanisms of the Ethereum platform to mitigate DDoS attacks. This is achieved by requiring payment for each transaction, thereby deterring attackers from inundating the system with excessive service requests. Ethereum's public Blockchain platform offers developers the flexibility to write and execute code, utilizing the concept of "gas" to manage transaction fees and resource allocation within the Ethereum virtual machine. Insufficient gas payment results in incomplete contract execution and the rollback of all changes. These solutions are divided into three categories: Solutions based on the Ethereum Platform with Traffic Control (SEPTC), Solutions based on the Ethereum Platform with Authorization (SEPA), and Solutions Simply based on the Ethereum Platform (SSEP).

2.1.1. SSEP Mechanism

This category of solutions uses the capabilities of the Ethereum platform to thwart DDoS attacks. They assert the resilience of the system against such attacks, attributing it to the distributed architecture and the transaction payment mechanism inherent in the Ethereum Blockchain, which deters excessive service requests. However, these solutions are still nascent and heavily reliant on the Ethereum platform for swift implementation. Moreover, they are limited to preventing attacks within Ethereum-based Blockchains and lack compatibility with other Blockchain platforms.

2.1.2. SEPTC Mechanism

The solutions proposed within this framework also emphasize traffic management to address DDoS attacks. This classification employs two specific strategies: setting transaction rate limits and implementing whitelisting mechanisms. To counter DDoS attacks, solutions in this category make use of both the Ethereum platform and traffic control strategies. However, integrating these technologies also presents a unique set of difficulties.

2.1.3. SEPA Mechanism

These solutions integrate the Ethereum platform with authorization mechanisms to counter DDoS attacks. However, the Ethereum platform lacks the capability to filter incoming devices, allowing malicious ones to exploit the platform freely. The primary issue with this approach is the absence of measures to detect and mitigate DDoS attacks. If an attacker circumvents the authentication process and initiates a DDoS attack, the system lacks effective mechanisms to identify and mitigate the attack.

3. FRAMEWORK FOR DDoS PREVENTION

The concept involves a multi-phase modular framework designed to act as a protective shield, defending applications against potential attacks akin to DDoS. Implementing this involves creating a decentralized application hosted on the web. Additionally, a safeguarding mechanism necessitates another blockchain application equipped with a deployed smart contract and a deep learning algorithm. The process entails employing smart contracts for initial traffic classification and implementing learning algorithms for traffic analysis.

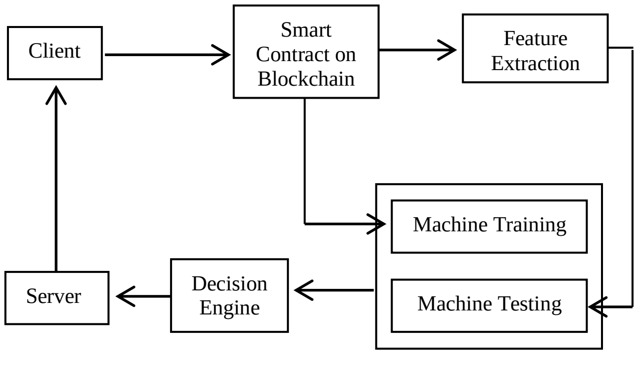

Fig. 2. Framework for DDoS Prevention

All incoming traffic directed towards a core blockchain-based application is rerouted to the blockchain (database), initiating an initial smart contract. This contract serves as a filter, categorizing requests based on predetermined parameters. This layer is essential to maintain the efficiency of the application's request-response cycle, ensuring it isn't impacted by the classifier's processing time. Requests meeting the initial criteria proceed directly to the server for processing, while any remaining suspicious requests are directed to the Machine Learning classifier for further evaluation. Requests flagged as suspicious, requiring thorough verification to ascertain whether they're associated with a Botnet, malicious intent, or a potential DoS attack, are directed to an ML classifier. This classifier utilizes deep learning techniques to assess if incoming traffic poses a threat. Verified safe traffic proceeds to the server for processing, while any identified as malicious are discarded, and their details are logged, storing the corresponding IP/MAC addresses.

The classifier's initial task involves real-time anomaly detection within the network. To achieve this, it studies the standard behavior of regular packets within the network. Real-time anomalies are monitored, and any suspicious packets are categorized accordingly. These packets undergo classification based on specific characteristics, including request type, source, accessed resource type, packet size, possible embedded scripts, etc. Blacklisted blocked packets are scrutinized by the smart contract and classifier, cross-referencing their details with the log file of identified malicious IP sources. This process significantly reduces the request response time, thereby enhancing usability while maintaining security measures.

4. PROPOSED SOLUTION

4.1. Dataset

One of the difficulties and Challenges associated with using machine learning is finding a large realistic training dataset. In this study, the general type of dataset is employed for the evaluation. By this dataset, we only define attacker and victim actual address, type of packets, number of packets deploying on victim machine. Here we use only 22 features of dataset, which are labelled as either normal or an attack as it is shown in Table 1. The four types of DDoS attacks included in the DDoS dataset are SIDDOS, HTTP-Flood, UDP-Flood, and Flood. A tiny percentage of the training and testing data is taken from this data set so that the model can be experimented with.

Table 2. Units for magnetic properties

Features	Description
1	S_ ADDR
2	D_ ADDR
3	HTTP_PKT _ID
4	FROM _NODE
5	TO _NODE
6	PKT _SIZE
7	FLAGS
8	FID
9	SEQ_ NUMBER
10	NUMBER _OF_ PKT
11	NUMBER _OF_ BYTE
12	NODE _NAME_ FROM
13	NODE _NAME _TO
14	HTTP_PKT _IN
15	HTTP_PKT_OUT
16	PKT _DELAY_ NODE
17	PKT_RATE
18	BYTE _RATE
19	PKT _SEND_ TIME
20	PKT _RESEVED _TIME
21	FIRST _PKT _SENT
22	LAST _PKT _REC

4.2. Decision Tree Pruning

Decision tree pruning is a new approach to eliminate the position of the decision tree into the specific state. With this help of Pruning, we can reduce the dimensions of decision trees by classify attacks. Pruning reduces the complexity of the ultimate classifier, and hence improves predictive accuracy by the reduction of over-fitting. Decision Tree Pruning is a technique to find a subset of the dataset from the original dataset. Alternately we can say that to eliminating features from the original dataset to obtain a subset of features that has higher accuracy on low-cardinality sets. It plays a key role in building detection models. However, In Table- I you can see that, It shows 22 features of the DDoS dataset, By the features of the decision tree algorithm, the full data set will reduce both the data and the computational complexity and improve both the efficiency and the exactness of the model. Here you can see that, using all 22 features without applying Decision Tree Pruning algorithm, It might increase the overhead of the model, which leads to increases in the time to build the model. The data must first be standardized before we can execute Decision Tree pruning and merge the standardized data into a single series of observations. The decision pruning algorithm feature that we put into practice automates this process by removing each feature from the entire collection of features and then determining if the subset of features is accurate. In the above table, we find only five features PKT _SIZE, SEQ_ NUMBER, PKT_RATE, BYTE _RATE, and LAST _PKT _REC from more features that are least significant. This process continues until no improvement of the accuracy is observed on the elimination of features. The pseudo-code is presented in the Algorithm that shows outline the steps of the method. The features given in Table 1 are the significant features obtained by the Decision Tree Feature Pruning algorithm used in our detection approach.

4.3. Algorithm of Decision Tree Pruning

```
Start
Decision_Tree_Pruning (A, K, Checked)
Input: A- Number of total required features
K- Number of N- decision tree vectors Checked- A set of combination of decision tree features which are already checked
Output:  R- Eliminated or reduced set of decisions (features) of A
if Checked carries (K) Then Return
end
if ( N-A=0) then
Return K, Find Accuracy () Else
Ai= (all decision tree vectors in K except decision tree i) for all i=1……N
Return Max
(Decision_Tree_Pruning ($A_1K_1$)
(Decision_Tree_Pruning ($A_2K_2$)
.....
Decision_Tree_Pruning ($A_NK_N$)))
// return subset with right accuracy
end
End
```

4.4 Initializing Markov Process with eliminating features

The further stage after getting the major features set and before training with the Markov process is to initialize the deleting features with parameters. In theory, one might randomly initialize the Markov parameters, and then use the Baum-Welch training algorithm (also called the Forward-Backward algorithm) to calculate or estimate them over a large number of training repetitions. To find the Markov parameters (transition probability matrix and emission probability matrix), one must begin with an imprecise estimate.

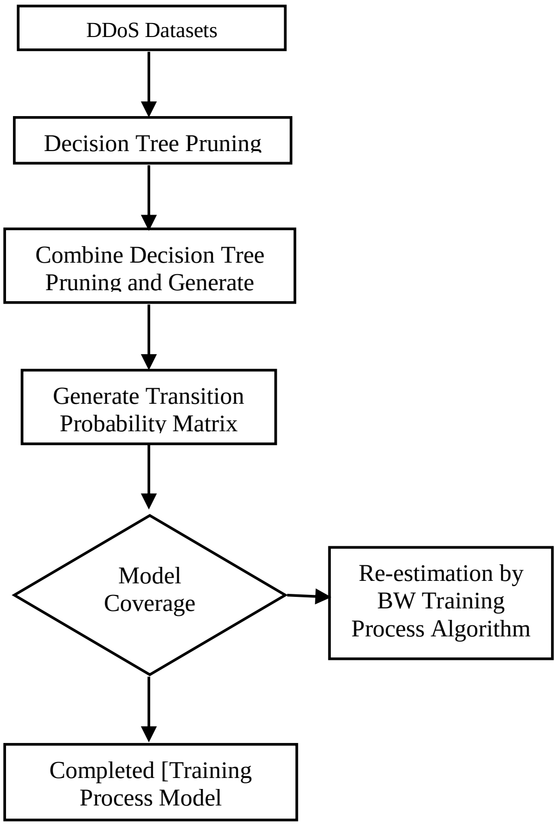
DDoS Datasets
Decision Tree Pruning
Combine Decision Tree Pruning and Generate
Generate Transition Probability Matrix
Model Coverage
Re-estimation by BW Training Process Algorithm
Completed [Training Process Model

Fig. 3. Flow of Markov Process for Elimination

Using Map- Reduce operation in cloud environment (Choi & Junho et al (2013), We can get rid of the dataset. After they settle, they can be assessed using the Baum-Welch algorithm to determine which parameters are more accurate and which Markov process better fits the observed sequence.

5. PERFORMANCE ANALYSIS AND EVALUATION

5.1. Performance Rate

After the applying decision tree pruning algorithm, We assess our model's performance rate as well as its accuracy in organizing and projecting the attack class label. The following four terms should be familiar to us: Positive Attack (True Positive): the quantity of attacks that are counted as attacks.

True Negative Attack (TNA): The quantity of non-attack cases that are categorized as such. False

Negative Attack (FNA): The quantity of attack cases that are categorized as non-attacks.

False Positive Attack (FPA): Table 2 displays the confusion matrix for a two-class example (attack and non-attacks) based on the number of non-attacks cases identified as attacks.

Table 2 Confusion matrix Attack and non-attacks

Two-Class Case		Predicted Class	
		Attack	Non-Attack
Actual Class	Attack	TPA	FNA
	Non-Attack	FPA	TNA

For this research work, the staging of the suggested model was tested for this study using the representation measures listed below: The ratio of all instances properly predicted to all instances is known as accuracy. In our research, the Viterbi algorithm is used to construct a likely state sequence, which is then compared against the known state sequence to determine TPA, FPA, FNA, and TNA. This is how accuracy is determined. The following formula can be used to determine the accuracy:

5.1.1. Accuracy

We apply the Viterbi algorithm method to find maximum accuracy. The Viterbi path is a dynamic programming approach that determines the most likely order of concealed attack states. the proportion of all instances that were accurately predicted to all instances overall. The Viterbi algorithm is used in our work to generate a likely state sequence, which is then compared to the known state sequence to determine TPA, FPA, FNA, and TNA accuracy measures. The provided equation can be used to calculate the correctness:

$$Accuracy=\frac{TPA+TNA}{TPA+TNA+FPA+FNA} \quad (1)$$

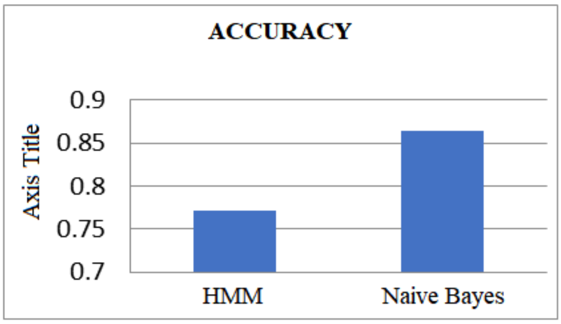

Fig. 4. Accuracy

5.1.2. Error Rate / False Negative Rate (FNR)

To find the Fake Negative Rate, we use the ratio of the total number of jumbled to total number of all predictions. To obtain TPA, FPA, FNA, and TNA, use this Viterbi technique to generate a likely state sequence and compare it against the known state sequence.

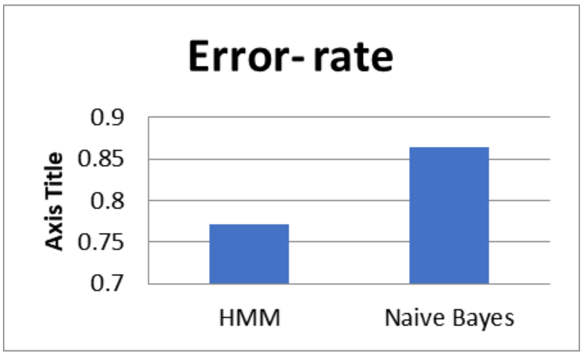

Figure 4: Error Rate

The error rate can be calculated by using the following equation:

$$Error\ rate = \frac{FPA+FNA}{TPA+FPA+TNA+FNA} \qquad (2)$$

5.1.3. Fall-out/ False Positive Rate (FPR)

The ratio of the total number of attack predictions to the number of detected fake positive attacks is used to calculate the False Positive Rate.

This can be calculated by using the given following equation:

$$Error\ rate = \frac{FPA}{FPA+TNA} \qquad (3)$$

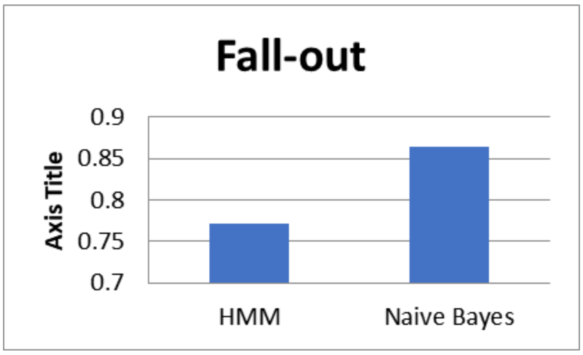

Figure 5. Fall Out

5.1.4. The sensitivity/ True Positive Rate (TPR)

We utilize the ratio of the total number of detected genuine positive attacks that are accurately identified as attacks to the total number of positive cases to determine the genuine Positive Rate.

The following formula can be used to determine the sensitivity:

$$\text{Sensitivity} = \frac{TPA}{P} \qquad (4)$$

Here, P= TPA+FPA.

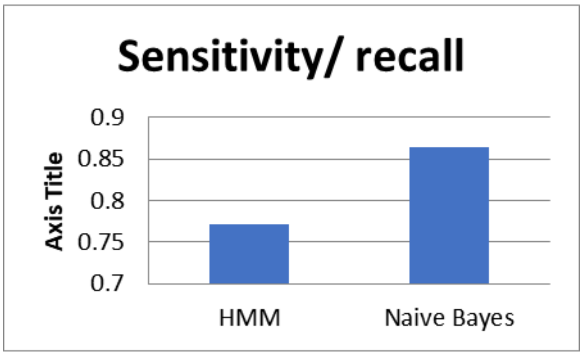

Figure 6 Sensitivity / Recall

5.1.5. The specificity/ True Negative Rate (TNR):

The proportion of all detected true negatives that are accurately classified as non-attacks to all negative occurrences.

The following formula can be used to determine the Specificity:

$$\text{Specificity} = \frac{TNA}{N} \qquad (4)$$

Here, N: The number of negative instances,

N=TNA + FNA

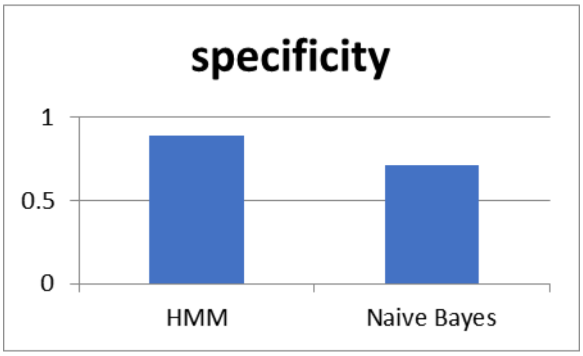

Figure 7. Specificity

5.1.6. F measure

By the F- measure test, we are testing the accuracy of the model and it taken into account both the recall and precision.

By using the following equation, F measure can be calculated-

$$\text{F } measure = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \qquad (6)$$

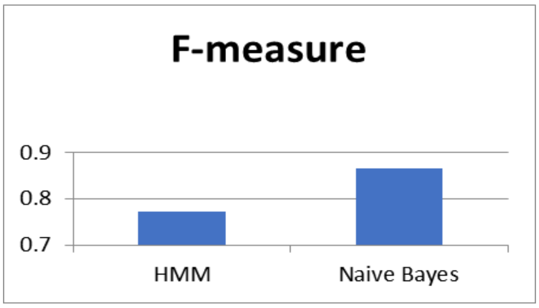

Figure 8. F-measure

6. RESULT AND ANALYSIS

The result shows our presented approach can gain better leads to terms of attack detection rate. Moreover, the result shows improved performance with a reduced feature set after applying the Feature Pruning algorithm and selected the most important features.

By testing it on the DDoS set and training an HMM, attacks were detected with higher than 97% reliability in most trial runs. . The representation of the HMM algorithm compares against classification algorithms of Naive Bayes algorithm. Data was taken from the WEKA. Figure: 9 shows a graphical way comparison between the two classification algorithms.

Table 3. The summarily of the experiment

Performance Measures/Classification Algorithm	Training/Testing	
	HMM	Naïve Bayes
Accuracy	0.8741	0.8348
Error Rate	0.0159	0.0552
Fall-Out	0.0102	0.0888
Sensitivity/ Recall	0.7413	0.8431
Specificity	0.8896	0.7112
F-Measure	0.7714	0.8644

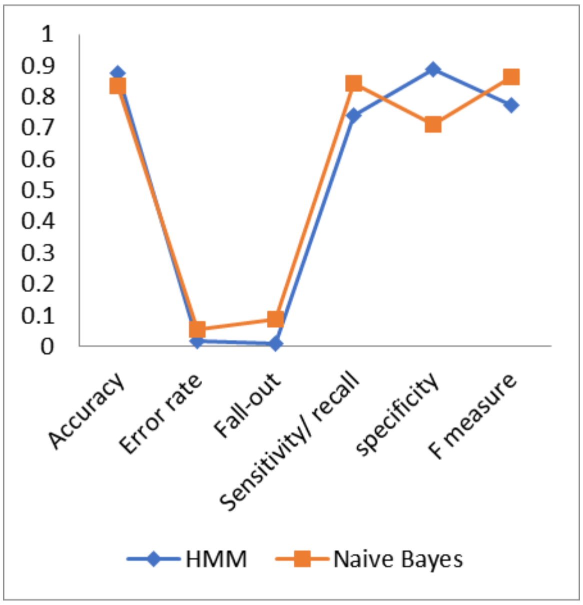

Figure 9: Graphical ways with the comparison of algorithms

7. CONCLUSION

In the present circumstances, machine-learning methods are benefitting the foremost scrutinization in forecast thanks to its capability to publish, develop, improve, and adjust. Thus, in this research, we demonstrated our detection approach using Hidden Markov Models (HMM) that applied and tested on the goggle DDoS dataset to detect the DDoS attack. By the above methodology, we can say that we are ready to produce an excellent presentation with high accuracy, low error rate, and Fake Positive Rate. The detection result demonstrated that Markov Process gives a more accurate result than would have been obtained by Naive Bayes algorithms while detecting the attacks. It achieved 96.21 % accuracy.

Acknowledgements

This research activity is conducted and supported Computer Center of CSJM University, Kanpur. I thank my supervisor Prof (Dr) Ajay Pratap who provided insight and expertise that greatly assisted the research. I also thank Prof. (Dr.) Harsh Dev for assistance with techniques, and methodologies.

Author contributions

Mr Himanshu Shukla: Software, Validation, Writing-Original draft, Visualization, Investigation and Editing

Prof. (Dr) Ajay Pratap: Conceptualization, Methodology, Software Selection

Prof (Dr) Harsh Dev: Review and Analysis of paper.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] A. Carlin, M. Hammoudeh, and O. Aldabbas, “Defence for Distributed Denial of Service Attacks in Cloud Computing,” Procedia Computer Science, vol. 73, pp. 490–497, 2015, doi: 10.1016/j.procs.2015.12.037.

[2] S. T.K and D. B, “Preventing Distributed Denial of Service Attacks in Cloud Environments,” International Journal of Information Technology, Control and Automation, vol. 6, no. 2, pp. 23–32, Apr. 2016, doi: 10.5121/ijitca.2016.6203.

[3] H. Abusaimeh, “Distributed Denial of Service Attacks in Cloud Computing,” International Journal of Advanced Computer Science and Applications, vol. 11, no. 6, 2020, doi: 10.14569/ijacsa.2020.0110621.

[4] A. Bonguet and M. Bellaiche, “A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing,” Future Internet, vol. 9, no. 3, p. 43, Aug. 2017, doi: 10.3390/fi9030043.

[5] A. O. Akinwumi, A. O. Akingbesote, O. O. Ajayi, and F. O. Aranuwa, “Detection of Distributed Denial of Service (DDoS) attacks using convolutional neural networks,” Nigerian Journal of Technology, vol. 41,

no. 6, pp. 1017–1024, Mar. 2023, doi: 10.4314/njt.v41i6.12.

[6] V. A. Koryakova, “Identifying Distributed Denial of Service Attacks,” Mathematical Methods in Technologies and Technics, no. 5, pp. 105–108, 2021, doi: 10.52348/2712-8873_mmtt_2021_5_105.

[7] “Retracted: Prevention Techniques against Distributed Denial of Service Attacks in Heterogeneous Networks: A Systematic Review,” Security and Communication Networks, vol. 2023, pp. 1–1, Dec. 2023, doi: 10.1155/2023/9805019.

[8] A. Bonguet and M. Bellaiche, “A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing,” Future Internet, vol. 9, no. 3, p. 43, Aug. 2017, doi: 10.3390/fi9030043.

[9] C. Vasan Sai Krishna, Y. Bhuvana, P. Pavan Kumar, and R. Murugan, “Reducing distributed denial of service (DDoS) attacks using client puzzle mechanism,” International Journal of Engineering & Technology, vol. 7, no. 1.1, p. 230, Dec. 2017, doi: 10.14419/ijet.v7i1.1.9473.

[10] “Defending Distributed Denial Of Service (Ddos) Attacks: Classification And Art,” Elementary Education Online, vol. 20, no. 02, Jun. 2021, doi: 10.17051/ilkonline.2021.02.266.

[11] S. SINGH, A. BHANDARI, K. K. SALUJA, and A. L. SANGAL, “Study to Validate the Performance of Flooding Based Distributed Denial of Service Attacks,” International Journal of Computer Networks and Communications Security, vol. 8, no. 1, pp. 1–9, Jan. 2020, doi: 10.47277/ijcncs/8(1)1.

[12] A. Nurdin and I. Riadi, “Network Forensic on Distributed Denial of Service Attacks using National Institute of Standards and Technology Method,” International Journal of Computer Applications, vol. 183, no. 40, pp. 39–47, Dec. 2021, doi: 10.5120/ijca2021921799.

[13] 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, Nov. 2019, doi: 10.1002/spy2.96.

[14] P. Krishna Kishore, S. Ramamoorthy, and V. N. Rajavarman, “ARTP: Anomaly based real time prevention of Distributed Denial of Service attacks on the web using machine learning approach,” International Journal of Intelligent Networks, vol. 4, pp. 38–45, 2023, doi: 10.1016/j.ijin.2022.12.001.

[15] Z. Shah, I. Ullah, H. Li, A. Levula, and K. Khurshid, “Blockchain Based Solutions to Mitigate Distributed Denial of Service (DDoS) Attacks in the Internet of Things (IoT): A Survey,” Sensors, vol. 22, no. 3, p. 1094, Jan. 2022, doi: 10.3390/s22031094.

[16] “DETECTION OF DISTRIBUTED DENIAL OF SERVICE ATTACKS IN SDN USING MACHINE LEARNING TECHNIQUES,” International Research Journal of Modernization in Engineering Technology and Science, Mar. 2023, Published, doi: 10.56726/irjmets34739.

[17] R. U and K. E, “Distributed Denial of Service Attacks Prevention, Detection and Mitigation – A Review,” SSRN Electronic Journal, 2021, Published, doi: 10.2139/ssrn.3852902.

[18] O. Tinubu, A. Sodiya, and O. Ojesanmi, “A behavioral model for characterizing flooding distributed denial of service attacks,” International Journal of Information Technology, vol. 15, no. 2, pp. 955–964, Sep. 2022, doi: 10.1007/s41870-022-01097-3.

[19] O. Tinubu, A. Sodiya, and O. Ojesanmi, “A behavioral model for characterizing flooding distributed denial of service attacks,” International Journal of Information Technology, vol. 15, no. 2, pp. 955–964, Sep. 2022, doi: 10.1007/s41870-022-01097-3.

[20] G. Spathoulas, N. Giachoudis, G.-P. Damiris, and G. Theodoridis, “Collaborative Blockchain-Based Detection of Distributed Denial of Service Attacks Based on Internet of Things Botnets,” Future Internet, vol. 11, no. 11, p. 226, Oct. 2019, doi: 10.3390/fi11110226.

[21] D. Patel*, “Blockchain Technology towards the Mitigation of Distributed Denial of Service Attacks,” International Journal of Recent Technology and Engineering (IJRTE), vol. 8, no. 6, pp. 961–965, Mar. 2020, doi: 10.35940/ijrte.f7420.038620.

[22] Dr. E. Punarselvam, “Blocking Distributed Denial of Service Flooding Attacks with Dynamic Path Detectors,” International Journal for Research in Applied Science and Engineering Technology, vol. 8, no. 6, pp. 1318–1322, Jun. 2020, doi: 10.22214/ijraset.2020.6212.