# Advanced Hybrid Anomaly Detection and Mitigation Framework for Slow HTTP DDoS Attacks

Ugwu Blessing Dominic*, A. E. Evwiekpaefe, M. E. Irhebhude and Aliyu Ahmed

**Abstract** This study investigates the Adaptive Hybrid Anomaly Detection and Mitigation (AHADM) framework's effectiveness against slow HTTP DDoS attacks. Comparing it to established methods, the research highlights AHADM's superior ability to swiftly detect and neutralize these elusive threats. By using adaptive algorithms and sophisticated anomaly detection mechanisms, AHADM outperforms existing solutions, offering robust defense against evolving cyber threats. The study also provides valuable insights into attack characteristics and practical implementation guidelines, contributing significantly to network security literature. AHADM emerges as a proactive defense strategy, mitigating risks posed by slow HTTP DDoS attacks in modern network environments.

**Key words:** AHADM Framework, Apache Hadoop, DDoS Attacks and MLP.

## 1 Introduction

The internet's exponential growth revolutionized global commerce, leading to increased cyber threats like Distributed Denial of Service (DDoS) attacks. Slow HTTP

Ugwu Blessing Dominic · A. E. Evwiekpaefe · M. E. Irhebhude
Dept. of Computer Science, Nigerian Defence Academy (NDA), Kaduna
e-mail: dominic.blessing@yahoo.co.uk

A. E. Evwiekpaefe
e-mail: aeevweikpaefe@nda.edu.ng

M. E. Irhebhude
e-mail: mirhebhude@nda.edu.ng

Aliyu Ahmed
Dept. of Computer Engineering, Federal University of Technology, Minna e-mail: aliyu.ahmed@futminna.edu.ng

DDoS attacks, particularly stealthy, operate slower over established connections, posing detection challenges [1, 2]. These attacks caused substantial financial losses across sectors and government bodies, coinciding with the rise of IoT-connected devices vulnerable to external control by malicious actors [3, 4].

Hadoop effectively addresses memory inefficiencies in traditional DDoS detection methods [5, 6]. However, application-layer DDoS attacks at OSI model's seventh layer present complexities and evade detection by targeting established connections [1]. Traditional methods suffer from slow responsiveness and struggle with large-scale data, relying on signatures and proving insufficient against evolving attack variants [7, 8, 9].

To address these issues, this study proposes the Apache Hadoop-based DDoS Detection and Mitigation (AHADM) framework. AHADM aims to detect and mitigate Slow Post, Slow Header, and Slow Read application layer DDoS attacks in near real-time by integrating the MLP algorithm and tailored threshold-based techniques for each attack type. The paper is organized as follows: Section 2 reviews related work in DDoS detection, Section 3 presents the research gap, while, the objectives and the hypothesis of the research are presented in Sections 4 and 5 respectively. The details of the methodology, expected results, justification, and validation of results are presented in Section 6, Section 7 discusses the expected contribution, and finally, Section 8 concludes the paper and outlines future directions for research and improvement.

## 2 Extensive Literature Survey

This section presents an in-depth background on network attacks, focusing on DDoS and application layer attacks. It delves into various big data technologies, explores the MLP algorithm, and reviews relevant literature on using big data for detecting and mitigating DDoS attacks.

### *2.1 Distributed Denial of Service (DDoS) Attack*

Unlike protocol and volumetric based DDoS attacks that exploit vulnerabilities in layers three and four, [7] classified Slow HTTP attacks as application Layer types that exploit layer seven protocols such as Slowloris (keeps numerous connections open with incomplete HTTP GET requests, consuming server resources and denying access to legitimate users [12, 13]), Slow HTTP POST (exploits web servers by sending oversized content length values, followed by slowly transmitted message body chunks, eventually leading to resource exhaustion [14]), and Slow Read [12, 13] (forces servers to wait for data updates by slowly reading server replies, causing denial of service through resource consumption via multiple connections).

Big Data, with its voluminous, diverse, and complex nature, demands specialized processing tools. Its prevalence in various fields raises concerns about data security amidst increasing DDoS attacks. Despite the potential benefits, challenges persist. Addressing these challenges is crucial to fully harness the potential of Big Data analytics, especially in enhancing security [16].

## 2.2 Deep Learning Algorithms

Deep learning algorithms, powered by Artificial Neural Networks (ANNs), autonomously extract data features without human input, adapting swiftly to different datasets [17]. Various algorithms like CNNs, LSTMs, RNNs, and more specialize in tasks like image recognition and sequential data analysis [18]. MLPs, a cornerstone of deep learning, learn both linear and non-linear functions while minimizing loss in training. Inspired by brain neurons, MLPs efficiently process inputs for predictive tasks [19]. These algorithms, including MLPs, contribute significantly to solving complex problems across diverse domains [18]. The functioning of the MLP involves a sequential process where data is initially fed into the input layer of the network [18]. Fig. 1 shows an MLP architecture.

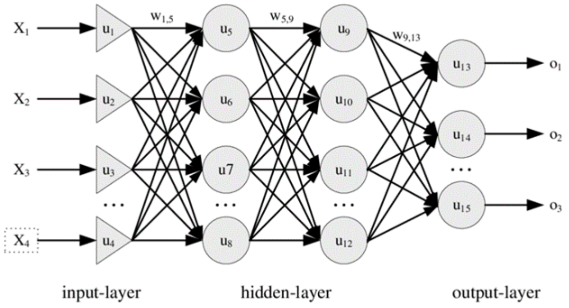

**Fig. 1** Architectural view of Multilayer Perceptron (MLP) (Wahyunggoro et al., 2013).

The $X_n$ are inputs into the input layer, the $U_n$ are the nodes ordered in the hidden layer, The $O_n$ are the outputs in the output layer and the $W_{mn}$ are the unidirectional connections with trainable weight for the MLP algorithm.

## 2.3 Review of Relevant Literature

Recent studies highlight the potency of advanced algorithms in fortifying security measures against DDoS attacks. [20] achieved a remarkable 98.99% detection efficiency using MLP deep learning algorithms against application layer DDoS attacks. [21] introduced the GHLBO algorithm, showcasing high accuracy in DDoS attack detection. [22] demonstrated the effectiveness of perplexity-based classification algorithms, boasting a 99% accuracy in detecting DDoS attacks in cloud computing environments. These studies underscore the efficacy of sophisticated algorithms in bolstering security measures.

A secure SaaS framework using Deep Belief Network (DBN) was presented in [23], achieving a noteworthy packet loss ratio of 16% while enhancing security protocols. Furthermore, Kumari and [24] compared Logistic Regression and naïve Bayes algorithms for DDoS attack detection, with Logistic Regression showcasing superior results. While, [9] focused on real-time prediction of application layer DDoS attacks, achieving a mean accuracy rate of 99.5%, setting the stage for proactive threat mitigation.

In parallel research, [25] explored time delay forecasting to detect slow HTTP DDoS attacks and recommended further investigations utilizing MLP algorithms for increased accuracy. Meanwhile, [26] proposed an attack-based filtering scheme that bolstered system efficiency by 12% in detecting HTTP slow rate DoS attacks, showcasing the potential for refined detection methodologies. While, [27] developed a combined Big Data and Machine Learning framework, demonstrating high accuracy in early DDoS attack detection.

Further research by [28] presented a software-based DoS protection device adept in analyzing, detecting, and mitigating cloud-based DoS attacks. A StackNet-based model was introduced in [29] showcasing a high detection accuracy of 99.3% across various application layer DoS attacks, emphasizing the importance of advanced detection techniques. In [30] a cloud-based intrusion detection system was designed, exhibiting high accuracy in real-time detection using Apache Spark and MLlib.

Collectively, these studies underscore the diversity of methodologies and frameworks in efficient DDoS attack detection and prevention. They emphasize the critical role of Big Data technologies, Machine Learning, and distributed computing in fortifying network security by offering effective strategies to combat evolving cyber threats.

## 3 Research Gap

The current DDoS attack detection literature primarily focuses on high-rate volumetric attacks in network and transport layers, particularly emphasizing HTTP get-flooded attacks. However, there's a research gap regarding slow HTTP application layer DDoS attacks. To bridge this gap, this study proposes an AHADM framework using the MLP algorithm and threshold-based techniques in the map reduce model for real-time detection and mitigation of these attacks. It aims to enhance accuracy and minimize complexity by focusing on vital flow properties. Building upon previous work by Savchenko and Hameed, this research aims to improve slow HTTP DDoS attack detection accuracy by considering unexplored forecasting intervals [25, 31]

## 4 Objectives

This study proposes the AHADM framework, leveraging Apache Hadoop, for real-time detection and mitigation of slow HTTP DDoS attacks. Specific bjectives include developing a robust MLP algorithm for detection, implementing non-complex threshold-based techniques for effective mitigation, evaluating performance in terms of accuracy and reduced complexity, and comparing it with existing DDoS approaches.

## 5 Proposed Hypothesis

The study aims to evaluate the AHADM framework's effectiveness in detecting and mitigating slow HTTP DDoS attacks compared to existing methods. It hypothesizes that AHADM will outperform current techniques by accurately distinguishing between genuine and attack traffic, reducing false positives, and demonstrating quicker response times. It is anticipated that the framework would adjust to changing attack tactics, strengthening defences against complex, slow HTTP DDoS attacks.

## 6 Proposed Methodology

This section gives an insight into the working model of the proposed AHADM framework. The methodology for this research is made up of the processes and procedures used to achieve the research aim and to provide answer to the research questions. The detailed process involves collection of live network packets without attack (legitimate packets) and packets with attack (attack packets) that will be set up in a docker container to create the datasets. The process will be composed of capturing of the network traffic packets, the collection and filtering of the traffic packets to generate traffic flows related to the attack types which will take place in the capturing server, then the flows will be transferred to the detection server where the detection of attack packets will take place with the implementation of the MLP algorithm and a non-complex threshold-based technique for the mitigation process. Once an attack packet is detected, result will be generated in order to implement the mitigation process to block or drop such packet and not allow it to pass through.

### *6.1 Research Design*

This research is aimed at developing an effective method for the detection and mitigation of slow HTTP application layer DDoS attacks by harnessing the capability of the MLP algorithm and a non-complex threshold-based technique to enhance the security and reliability of the Hadoop system in the AHADM framework. This entails collecting network packets, filtering them to produce traffic flows linked to attacks on a capturing server, moving these flows to a detection server for the implementation of the MLP algorithm, and using a threshold-based method for detection and mitigation. Attack packets are dropped or blocked by the mitigation procedure when they are identified.

## 6.2 The Architecture of the Proposed AHADM Framework

The study presents AHADM (Apache Hadoop-based DDoS Detection and Mitigation) to identify and counter slow HTTP DDoS attacks in real-time. AHADM leverages Apache Hadoop and the MLP algorithm along with a basic threshold-based method. The process begins by capturing network traffic using Tshark within the AHADM framework, storing parameters and selected features in Hadoop's Distributed File System (HDFS). Detection happens on a dedicated server, employing the Map Reduce programming model and MLP algorithm through the YARN resource allocation framework. The framework's architecture is depicted in Fig. 2.

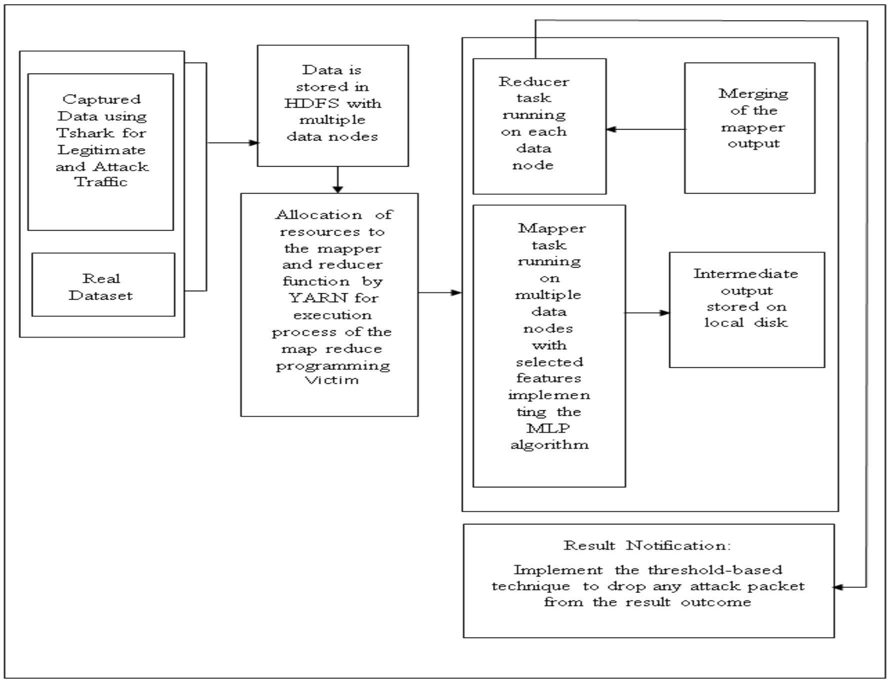
Captured Data using Tshark for Legitimate and Attack Traffic
Real Dataset
Data is stored in HDFS with multiple data nodes
Allocation of resources to the mapper and reducer function by YARN for execution process of the map reduce programming Victim
Reducer task running on each data node
Merging of the mapper output
Mapper task running on multiple data nodes with selected features implementing the MLP algorithm
Intermediate output stored on local disk
Result Notification:
Implement the threshold-based technique to drop any attack packet from the result outcome

**Fig. 2** The architecture of the proposed AHADM framework.

### *6.3 Design framework of the Study*

The AHADM framework aims to swiftly detect and counter slow HTTP DDoS attacks in near real-time. Tshark will facilitate log transfer from the capturing server to the detection server. In the data collection phase, an HTTP traffic dataset will be curated in a Hadoop environment, encompassing normal traffic and variations of slow HTTP DDoS attacks. This dataset is pivotal for MLP algorithm training. After preprocessing, including feature normalization and handling missing values, the MLP algorithm will be trained using known attack patterns. The trained model will then be tested on a separate dataset to assess accuracy and performance (see Table 1 for selected features), while, the algorithm for Slow header, Slow message body and Slow read HTTP DDoS Attacks using MLP algorithm is presented in Algorithm 1. Meanwhile, Fig. 3 depict the work flow diagram showing each of the design process for the detection and mitigation process.

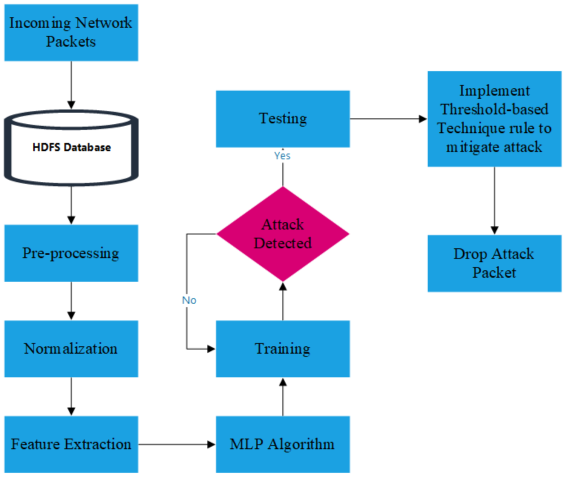

**Fig. 3** Workflow diagram of the detection and mitigation process in the AHADM framework.

**Table 1** Selected Features for MLP algorithm and their description

| S/N | Features | Description |
|---|---|---|
| 1. | flowDur | Flow Duration |
| 2. | noOfComm | Number of established connections by clients |
| 3. | SerTime | Total service time of a client |
| 4. | noOfkeepalive | Number of urgent keep-alive messages |
| 5. | IncompWaitTime | Waiting time to receive remaining request |
| 6. | avgPktSize | Avg pkt size of urgent data/keep-alive msgs in a flow |
| 7. | avgGetPostIntv | Avg time interval between successive GET/POST req in a flow |
| 8. | SrcIP | Source IP address |
| 9. | SrcPort | Source Port |
| 10. | IncompReqIntv | Time Interval between partial request (GET/POST) in a flow |
| 11. | noOfIncompReq | Number of times partial request are been communicated in a flow |
| 12. | noOfGetPost | Number of HTTP GET/POST request in a flow. |

**Algorithm 1:** Slow Header, Msg Body, and Read Attacks Detection

```
Input  : $T \leftarrow [T_1, T_2, \ldots, T_m]$: Represents incoming network traffic
Output: Detected slow header, message body, and read attacks
Create Network flows for each incoming traffic: $f \leftarrow [f_1, f_2, \ldots, f_n]$;
foreach $f_i \in [f_1, f_2, \ldots, f_n]$ do
    $flowDur = ComputeFlowDuration(f_i)$;
    $incompleteRequest = FindIncompleteRequest(f_i)$;
    $noOfIncomReq = Count(incompleteRequests)$;
    $IncompwaitTime = CompIncompReqWaitTime(incompReqs)$;
    $connectionsByClient = CountConnectionsByClient(f_i)$;
    $noOfkeepalive = ComputeNoOfKeepAliveRequests(f_i)$;
    $avgPktSize = ComputeAveragePacketSize(f_i)$;
    $avgDataIntv = ComputeAverageDataInterval(f_i)$;
    $avgIncomreqIntv = CompAvgIncompReqInterval(f_i)$;
    $noOfGetPost = CountGetPostRequests(f_i)$;
    $avgGetPostIntv = ComputeAverageGetPostRequestInterval(f_i)$;
    if $IsSlowHeaderAttack(f_i, noOfConn, flowDur, IncompWaitTime)$ then
        Declare $f_i$ as a slow header attack;
    end
    if $IsSlowMessageBodyAttack(f_i, noOfConn, flowDur, noOfkeepalive,$
    $avgPktSize, avgDataIntv, noOfIncomReq, avgIncomreqIntv)$ then
        Declare $f_i$ as a slow message body attack;
    end
    if $IsSlowReadAttack(f_i, noOfConn, flowDur, noOfGetPost, avgGetPostIntv)$ then
        Declare $f_i$ as a slow read attack;
    end
end
```

The algorithm introduces IsSlowHeaderAttack, IsSlowMessageBodyAttack, and IsSlowReadAttack functions to encapsulate conditions for each attack type, enhancing readability and modularity. These functions check specific attack-related conditions and return Boolean values, indicating the flow's attack type. This approach ensures a structured and readable format, improving the algorithm's clarity and maintainability.

### 6.3.1 A Non-Complex Threshold-based Technique

A non-complex threshold-based technique will be employed to define specific thresholds for different features that will be considered indicators of slow HTTP application layer DDoS attacks. The threshold will be determined based on non-complex rules and expert knowledge. Any deviation from the set threshold is considered as a potential attack. The following configuration settings is set for the threshold-based technique: IP address entry rate per second, N, file size entry, F, IP address threshold entry rate, and Threshold file size, $F_T$. The threshold-based technique is represented by:

$$statsus_{ID} = \begin{cases} IP_{attack},\ if\,\beta(N,F)>\beta(N_T,F_T) \\ IP_{legit},\ \ otherwise \end{cases}, \tag{1}$$

where $status_{ID}$ represent the IP address status, $IP_{attack}$ represents the IP address marked as attack IP, and $IP_{legit}$ represents the IP address marked as legitimate IP. Both the $\beta(N,F)$ and $\beta(N_T,F_T)$ are functions for the normal and threshold entry parameters respectively. The flow duration ($F_D$) and average time interval ($A_{VT}$) will establish threshold values by computing their averages. These thresholds will serve as the basis for implementing the technique under specific conditions.

## *6.4 Implementation Methodology*

The AHADM Framework implementation unfolds through five strategic phases. The Capturing Phase on a Windows victim machine involves Tshark for capturing and transforming raw network traffic data into log files. In the Attacker Phase, multiple virtual machines generate three types of Slow HTTP DDoS attacks ('Slow Header,' 'Slow Post,' and 'Slow Read'), targeting the victim's Apache web-server. The Detection and Analysis Phase employs a Hadoop cluster with MapReduce and the MLP algorithm to process log files in parallel.

While, the mitigation phase will receive the analyzed log file, evaluates the IP addresses' status ID, filters the attacker IP addresses; and send the configuration changes to the Apache webserver. The configuration settings will include denial of access settings for the attacker IP addresses based on the mitigation rule of packet rate and the absolute time out connection for each packet. This information will then be stored on the Apache webserver and will be used to drop or block the attacker's marked IP addresses.

Lastly, the result generation phase will interact with both the capturing and detection phases to collect the results of the detection and mitigation process. These results forms the bases to be use to carry out the performance evaluation of the proposed framework.

## *6.5 Proposed Results, Justification, and Validation*

### 6.5.1 Results

The results envisaged from the implementation of this research will involve implementation and deployment of the AHADM framework capable of accurately detecting and mitigating against slow HTTP DDoS attacks with superior performance against the state-of-the-art approaches. Most importantly, we will consider response time, detection accuracy, robustness, adaptability, and false positive rates as the main metrics for performance benchmark against the state-of-the-art approaches for combating slow HTTP DDoS attacks. We envisaged that, by integrating our threshold-based algorithm with the MLP algorithm, the framework will potentially perform better compared to the other approaches.

### 6.5.2 Justification

The proposed AHADM framework is dependent on a distributed architecture to detect and mitigate the slow HTTP DDoS application layer attacks using low-cost commodity hardware, which can help government agencies, organizations and educational institutions to secure their network in-house at a minimum possible cost and response time. The results obtained from this research will determine the efficiency rate of the apache Hadoop big data technology framework used in the detection and mitigation of the stated application-layer DDoS attacks on a network in terms of speed and accuracy rate.

The successful completion of this research will contribute to the field of network security by developing an advanced DDoS detection and mitigation system. The proposed framework has the potential to significantly improve the accuracy and efficiency of DDoS detection and mitigation, enabling prompt and effective measures. This research will benefit government agencies, organizations, educational institutions, financial institutions and small and medium enterprises to set up a low-cost in-house defence and, most importantly, save the enormous cost incurred when deploying a third-party DDoS mitigation service provider by providing them with enhanced protection against DDoS attacks.

### 6.5.3 Validation

The results of the AHADM framework will be validated using various experimentation and thorough evaluation based on different scenarios. State-of-the-art Statistical, machine and deep learning based approaches will be used for benchmarking the accuracy of the results. This will involve training the MLP algorithm, testing with various datasets, including regular traffics and compromised traffic with slow HTTP application layer DDoS attacks. The performance metrics to be considered are: response time, detection accuracy, robustness, adaptability, and false positive rates.

## 7 Expected Contribution to the Literature

This research is expected to contribute immensely to the body of knowledge through:

1. New evaluation methodologies that suit slow HTTP DDoS attacks and other application layer attacks.
2. Generation of datasets that would be useable to other researchers and approaches.
3. Improved detection and mitigation technique for combatting slow HTTP DDoS attacks and overall improvement in providing a secured cyber space.

## 8 Conclusion

The AHADM framework for slow HTTP DDoS attacks is presented in this paper. It detailed the concept and approach that involves the integration of threshold-based algorithm with MLP algorithm to optimize detection accuracy and reduces time complexity. The framework is based on the MapReduce model and Hadoop apache server. The implementation is based on multilayer approach to create robustness, reliability, and efficiency. The phases include capturing, attacking, detection and analysis, mitigation, and results generation. Testing and thorough evaluation against the state-of-the-art using response time, detection accuracy, robustness, adaptability, and false positive rates, would provide the bases for benchmarking.

## References

1. Punitha, V., Mala, C., & Rajagopalan, N.: A Novel Deep learning Model for Detection of Denial of Service Attacks in HTTP Traffic over Internet. International Journal of Adhoc and Ubiquitous Computing. **33**(40), 240–256 (2020)

2. Akinwunmi, A.O., Akingbesote, A.O., Ajayi, O.O., & Aranuwa. F.O.: Detection of Distributed Denial of Service (DDoS) Attacks using Convolutional Neural Networks. Nigerian Journal of Technology (NJOTECH). **41**(6), 1017–1024 (2022)
3. Zargar, S. T., Joshi, J., & Tipper, D.: A survey of defense mechanisms against Distributed Denial of Service (DDoS) flooding attacks. IEEE Communications Surveys and Tutorials. **15**(4), 2046?-2069 (2013)
4. Alese, T., Owolafe, O., Thompson, A.F., & Alese, B.K.: A User Identity Management System for Cybercrime Control. Nigerian Journal of Technology (NJOTECH). **40**(1), 129–139 (2021)
5. Ibor, A.E.: Zero day exploits and national readiness for cyber-warfare. Nigerian Journal of Technology. **36**(4), 11–74 (2018)
6. Najafabadi, M.M.: Machine Learning Algorithms for the Analysis and Detection of Network Attacks. Florida Atlantic University. ProQuest Dissertations Publishing, (2017)
7. Prasad, M.D., Babu, P., & Amarnath, C.: Machine Learning DDoS Detection Using Stochastic Gradient Boosting. International Journal of Computer Sciences and Engineering. **7**(4), 157–166 (2019)
8. Patil, N. V., Rama, C. K., Krishan, K., & Sunny, B.: E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks, Journal of King Saud University ? Computer and Information Sciences. 1–15 (2019)
9. Awan, M.J., Farooq, U., Babar, H.M.A., Yasin, A., Nobanee, H., Hussain, M., Hakeem, O., & Zain, A.M.: Real-Time DDoS Attack Detection System Using Big Data Approach. MDPI Journal. **13**(10743), 1–19 (2021)
10. Mangrulkar, N.S., Patil, A. R., & Pande, A.S.: Attacks and their Detection Mechanisms: A Review. International Journal of Computer Applications. **90**(9), 36–39.
11. Alomari, E., Manickam, S., Gupta, B.B., Karuppayah, S., & Alfaris, R.: Botnet-based Distributed Denial of Service (DDoS) Attacks on Web Servers: Classification and Art. International Journal of Computer Applications. **49**(7), 24–32 (2012)
12. Cambiaso, E., Papaleo, G., Chiola, G., & Aiello, M.: Slow DoS attacks: definition and categorization. International Journal of Trust Management in Computing and Communications. **1**(3-4), 300–319 (2013)
13. Choi, J., Choi, C., Ko, B., & Kim, P.: A Method of DDoS Attack Detection using HTTP Packet Pattern and Rule Engine in Cloud Computing Environment, Journal of Soft Computing, **18**(9), 1697?-1703 (2014)
14. Dantas, Y. G., Nigam, V., & Fonseca, I. E.: A Selective Defense for Application Layer DDoS Attacks. In: Proceedings of the 2014 IEEE Joint Intelligence and Security Informatics Conference (JISIC), the Hague, Netherlands, 75–82 (2014)
15. Damon, E., Dale, J., Laron, E., Mache, J., Land, N., & Weiss, R.: Hands-on denial of service lab exercises using Slowloris and RUDY. In: Proceedings of the 2012 Information Security Curriculum Development Conference, Kennesaw, Georgia, 21-29 (2012)
16. Cardenas, A.A., Manadhata, P.K., & Rajan, S. P.: Big Data Analytics for Security Intelligence. University of Texas at Dallas Cloud Security Alliance. 5, 1–22 (2013).
17. Goodfellow, I., Bengio, Y., & Courville, A.: Deep Learning. The MIT Press (2016)
18. Biswal, A.: Blog Post on The Top Ten Deep Learning Algorithms, https://www.simplilearn.com/tutorials/deep-learning-tutorial/deep-learning-algorithm, Last updated on 2023/08/29
19. Vasou, M.J., Ataie, E., & Bastam, M.: An MLP-based Deep Learning Approach for Detecting DDoS Attacks. Tabriz Journal of Electrical Engineering (TJEE). **52**(3), (2022)
20. Ahmed, S., Khan, Z. A., Mohsin, S.M., Latif, S., Aslam, S., Mujlid, H., Adil, M. & Najam, Z.: Effective and Efficient DDoS Attack Detection using Deep Learning Algorithm, Multi-Layer Perceptron. Journal of Future Internet. **15**(76), 1–24 (2023)
21. Balasubramaniam, S., Joe, C.V., Sivakumar, T.A., Prasanth, K., Kumar, S., Kavitha, V., & Dhanaraj, R.K.: Optimization Enabled Deep Learning-Based DDoS Attack Detection in Cloud Computing. International Journal of Intelligent Systems. **2023**(2039217), 1–16 (2023)
22. Mishra, N., Singh, R. k., & Yadav, S. k.: Detection of DDoS Vulnerability in Cloud Computing using the Perplexed Bayes Classifier. Journal of Computational Intelligence and Neuroscience. **2022**(151847), 1–13 (2022)

23. Theja, S., & Shyam, G.: A machine learning based attack detection and mitigation using a secure SaaS framework. Journal of King Saud University ? Computer and Information Sciences. **34**(7), 4047–4061 (2022)
24. Kumari, K., & Mrunalini, M.: Detecting Denial of Service attacks using machine learning algorithms. Journal of Big Data. 9(56), 4047–4061 (2022)
25. Savchenko, Vitalii.: Detection of Slow DDoS Attacks based on User?s Behavior Forecasting. International Journal of Emerging Trends in Engineering Research. **8**. 2019–2025 (2020)
26. Gutierrez, J.N., & Lee, K.: An Attack-based Filtering Scheme for Slow Rate Denial-of- Service Attack Detection in Cloud Environment. Journal of Multimedia Information System. **7**(2), 125-=136 (2020)
27. Shendi, M.M., Elkadi, H.M., & Khafagy, M.H.: Real-Time Attacks Detection Model and Platform Using Big Data and Machine Learning. International Journal of Scientific & Technology Research. **9**(9), 108-116 (2020)
28. Rahman, M.U., & Srinivasan, M.: A Critical Analysis of Denial of Service (Internet & Network Service) Attacks and their Detection. Journal of Critical Review. **7**(12), 3632–3639 (2020)
29. Smadi, S., Alauthman, M., Almomani, O., Saaidah, A., & Alzobi, F.: Application Layer Denial of Services Attack Detection Based on StackNet. International Journal of Advanced Trends in Computer Science and Engineering. **9**(3), 3929–3936 (2020)
30. Mounir, H., & Farah, J.: Comparative Study between Big Data Analysis Techniques in Intrusion Detection. Journal of Big Data and Cognitive Computing. **3**(1), 1–13 (2019)
31. Hameed, S., & Ali, U.: HADEC: Hadoop-Based Live DDoS Detection Framework. EURASIP Journal of Information Security. **11**, 1–19 (2018)