Arshad et al. LGURJCSIT 2024
ISSN: 2521-0122 (Online)
ISSN: 2519-7991 (Print)

LGU Research Journal of Computer Science & IT
doi: 10.54692/lgurjcsit.2024.081481
Vol (8): Issue (1), January – March 2024

Countering DDoS threats: leveraging ensemble methods for detection and mitigation

Muhammad Zeeshan Arshad[1], Zafar Iqbal[2*], Syed Haleemullah Shah[3]
Deptartement of Cyber Security, Air University, Islamabad, Pakistan.

Email: zafar.iqbal@mail.au.edu.pk

ABSTRACT:

Cloud computing is the modern concept of distributing numerous services through the Internet, such as web applications, databases, and individuals that operate on several servers. As cloud computing technologies advance, there is a growing risk of cyberattacks that can lead to service interruptions when storing and transmitting data. The most common sort of attacks against Cloud settings is distributed denial-of-service (DDoS). Several approaches for detecting and mitigating these attacks have been offered, they are ineffective because they still fail to fully protect the system when a newer type of attacking strategy is used against the systems. In this research, we propose a method for detecting and mitigating DDoS attacks in their early phases, considering top-layer advances at the application layer and the TCP handshake mechanism. This study employs a variety of ensemble-based machine learning approaches to classify incoming data as legitimate or malicious to respond to DDoS attacks at the application layer. Furthermore, the double TCP connection concept is used to prevent DDoS. Experiments show that the stacked voting system detects DDOS attacks with the best F-score of 99.9%.

KEYWORDS: Machine Learning, Cloud Computing, Distributed Denial-of-Service (DDoS), Double Transmission Control Protocol (TCP), and Cyber Attacks.

1. INTRODUCTION

Cloud computing provides its customers with a variety of services, such as platform as a service (PAAS), infrastructure as a service (IAAS), and software as a service (SAAS). Accordingly, many organizations have moved their workloads to cloud environments. While the cloud is renowned for its business drivers, such as Agility and Continuity, their popularity makes them prime targets for cyber attackers [1] [2]. For example, an attacker sends too many requests to a server. As a result, the server cannot handle such many requests simultaneously, which causes a denial of service (DoS). So, it goes down, becomes unavailable, or its services becomes degraded to their legitimate users [3]. DDoS attacks were the fifth most prominent among cloud-based attacks in 2013, according to the Cloud Security Alliance [4]. Cloud world, a leading technology analysis organization, highlighted DDoS as a major concern in cloud security, impacting the cloud's perceived reliability [5]. A record-breaking DDoS attack occurred between March 19th and March 23rd, 2013, with approximately 300 Gbps of incoming traffic observed [6]. Cloudflare, a well-known website security provider [7], successfully mitigated the attack and prevented a denial-of-service situation. Previously BI-LSTM-GMM model [8] has been proposed, in which the incoming traffic is pre-processed with the historical traffic and then passed to the Machine learning BiLSTM [9] model. Subsequently, upon the prediction of the Bi-LSTM model, traffic is labeled as malicious or legitimate traffic. However, this model is computationally expensive. The significant contribution of this paper is to address DDoS attacks, both on the application layer as well as the transport layer. The transport layer attacks are detected and prevented by integrating a Double TCP-enhanced handshake [10], which tries to ensure that only legitimate traffic will come to the server or application. In brief, this module works on the defined threshold value of the incoming traffic. If the value exceeds the defined limit, this module triggers and establishes a client-server connection using the five packets instead of the customary three-way handshake. In contrast to transport layer DDoS attack, the application layer attack is more dangerous to a server because it consumes the application connection resources and makes it unresponsive to legitimate users. This research has proposed a Behavior Based Malicious Request Detection (BBMRD) to handle this attack. BBMRD is a machine- learning model that classifies traffic and labels it benign or malicious. The rest of the paper is as structured as follows. Sections II provides the background knowledge. The literature review is presented in Section III. Then, the proposed method is described in Section IV. Implementation details are provided in Section V. Afterwards, an evaluation of the proposed method is shared in Section VI. Subsequently, results are discussed in Section VII. Finally, the conclusion and Future work are provided in Section VIII. II. BAC

2. BACKGROUND

Generally, DDoS attacks are classified into two major types. One type targets the transport Layer, i.e., the third layer of the TCP/IP model. While the other type of DDoS attack targets the application layer. Details of these are provided below:

2.1. Transport Layer Attack

In this, an attacker floods the network with

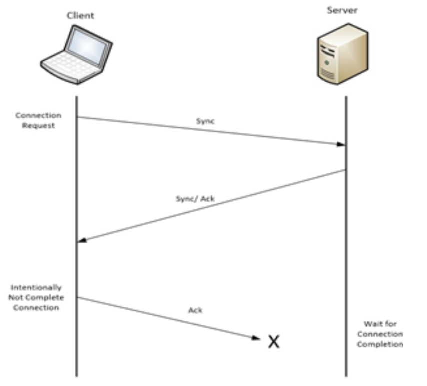
Client
Server
Connection Request
Sync
Sync/ Ack
Intentionally Not Complete Connection
Ack
X
Wait for Connection Completion

Figure 1: Half-Open TCP Connection [11]

connection requests that are never completed, called half-open connections [11]. These requests occupy the server's resources on which the attack is being generated but do not directly affect the client systems. A diagrammatic representation of the TCP half open connection is shown in Figure 1.

2.2. Application Layer Attack

The application layer is the TCP/ IP architecture's [12] top shell that links directly to the end systems. Now, as the cloud provides many services, it is possible to attack the application layer (the uppermost layer of the OSI model). To perform DDoS on any system, the application-layer attackers need limited resources and minimal bandwidth. The primary purpose of flooding the application layer is not a server-provided service but to consume all connections in parallel so that the server cannot allow any newer connections. For example, an attacker targets the application by sending the HTTP and FTP requests which could consume the computational limit Servers have a group of listeners that handle incoming requests at the same time. In these kind of attacks, attackers flood these listeners with non-stop requests. These types of attacks are known as Slow HTTP post attacks [12].

3. LITERATURE REVIEW

Christian Esposito et al. proposed the methodology of an interconnected federated cloud that provides low computational cost and security from DDoS attacks [13]. Krishna Modi et al. shared a new detection and prevention system with the help of hidden Markow model-based architecture. This model was developed for the detection and prevention of DDoS attacks. The authors devised the idea to avoid malicious packets and only allow legitimate source data/packets to pass. This technique helped them to avoid DDoS at the transport layer [10]. Jahanzaib et al. evaluated publicly accessible web vulnerability scanners against the top ten OWASP vulnerabilities in their article [14]. Furthermore, researchers proposed a tool for assessing the security of web applications. Shamshair et al. conducted an in-depth review of well-known algorithms for machine and deep learning, as well as the difficulties they confront in detecting zero-day attacks [15]. Furthermore, they provided the comparative evaluation results for the datasets with the best accuracy, precision, recall, and F1 score. Roshan Srivastava et al. shared various DDoS attack detection strategies. They compared and analyzed the techniques proposed to prevent DDoS attacks on cloud systems [16]. Massimo Ficco and his team proposed an inter-federated cloud using the Publish-subscribe service. As in Cloud Federation, several resources from independent clouds are combined to form a single cluster. This type of cloud facilitates users a method for various methods and overcomes vertical and horizontal scaling with minimum cost . S.N. Panda and his team proposed a method for a DDoS attack by using the concept of a threshold-based approach. The presented method analyzed every packet entering the network traffic based on their abnormal behaviors of the threshold value. Panda et al. proposed a technique based on Multiple Dynamic Thresholds to detect various DDoS attacks 3 [17]. Saravana Kumar et al. shared a method for preventing DDoS attacks on Cloud systems. They came up with the idea of using dynamic attacks cation of resources to prevent DDoS attacks on the cloud against a single user [18]. Fenil Kattacksla et al. analyzed different methods for detection and overcoming DDoS attacks using different techniques and tools [19]. These earlier methods were ineffective, especially when dealing with more recent kinds of attacking strategies. We established Long Short Term Memory (LSTM) models in order to address this challenge. This strategic addition takes place at the time of the client-server's first connection and is intended to mitigate attacks that are hidden in the Application layer—such as those that are included inside HTTPS or XML requests. LSTM integration's main job is to carefully examine every request that comes in and make sure it's secure before allowing it to be executed. This approach functions as a strong tool for locating and addressing possible application-layer-specific vulnerabilities. The system's overall security is greatly strengthened by the addition of LSTM, which offers a more robust protection against malicious activity and makes the establishment of a secure communication channel between the client and server.

4. PROPOSED METHOD

In the context of computer networking and communication protocols, SYN (Synchronize) and ACK (Acknowledgment) are fundamental components of the TCP (Transmission Control Protocol) three-way handshake. When two devices, such as a client and a server, initiate a connection, the process begins with the client sending a SYN packet to the server, indicating its intent to establish a connection. The server, upon receiving the SYN packet, responds with an ACK packet to acknowledge the receipt of the SYN and communicates its readiness to establish a connection. The client, in turn, acknowledges the server's response with another ACK packet. This three-step process ensures a synchronized connection setup between the client and server. This paper proposes double TCP and ensemble techniques to detect and prevent DDoS attacks at the transport and application layers. Double TCP SYN is employed to make a communication bridge in the middle of the server and clients to ensure the legitimate client generates the request. This technique prevents DDoS attacks because the system will not establish a connection between server and client unless the client does not Authors/ Mechatronics, Electrical Power, and Vehicular Technology XX(20XX) XX-XX send replies to the server with patterns to authenticate itself. While the connection is established between the client and server, there is still the possibility of attacks hidden in the Application layer (HTTPS or XML requests). For this purpose, we introduced Long Short Term Memory (LSTM) model, which will further check the request and then safely send it forward for execution. Details of these techniques are provided below.

4.1. Double TCP Connection

In this section, we acknowledge the foundation laid by Krishna et al. [10] in introducing the double TCP connection mechanism. While inspired by their work, our research extends and applies this concept in a distinct context, providing additional insights and modifications tailored to the specific requirements of our study. Specifically, we adapt the double TCP connection to the existing architecture of the TCP protocol, with a focus on server-side modifications. Contrary to its name, this method uses more than one TCP connection to communicate through the ends. A traditional TCP connection, a 3-way handshake, is used to establish a connection between server and client, using only 3 phases. A double TCP connection [10] uses five packets to establish a usual TCP connection, as seen in Figure 2. Firstly, a 3-wayhandshake intentionally keeps the TCP connection incomplete for the server. Next, a second 3-way handshake overwrites the previous ly incomplete TCP connection using the same source TCP port number and I.P. address but with a new identity field. It uses the SYN Cache to store the TCP connection protocol. The double TCP connection mechanism is well-suited to the existing architecture of TCP protocol. On the client side, a Double TCP connection works like a regular TCP connection; only server-side modification is required to implement this mechanism. Figure 1 explains the detailed mechanism. The client initiates the connection just like a standard mechanism; a server at its end catches the request and responds with the SYN packet, then the client again generates the ACK. The server, on purpose, holds the connection; in this case, if the client is legitimate, it will receive the packets. The client may or may not send a final acknowledgment that the server ignores. This time, the client starts the TCP connection by reestablishing the connection by sending an SYN packet. Still, the client sends an Identity field containing the 16-bit pattern received in the previous SYN-ACK packet from the server. The server gets the source I.P., port number, and Identity field from packets. The server will now check the entry in the connection request table. If the entry existed in the Table, the source would respond with SYN-ACK. Else the packet is dropped. The client must respond with the final ACK on the SYN-ACK received from the server, and the connection is established successfully. By using this method, there will be no half open TCP connection will be established on the server side. Upon successfully establishing the connection, the connection will be handled by the BBMRD to check the malicious requests that intend to attack the application.

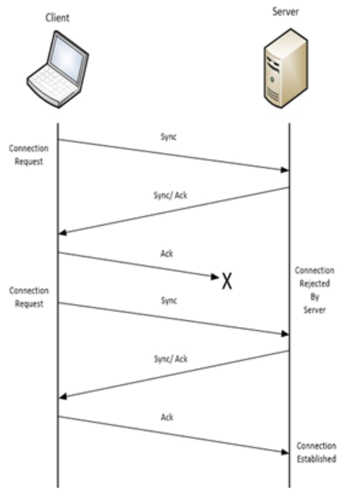

Figure 2: Double TCP Connection [10]

4.2. Behavior Based Malicious Request Detection (BBMRD)

A system entitled Behavior Based Malicious Request Detection, or BBMRD, is used in the context of the study that is being presented. It examines incoming packets and classifies them as either legitimate or malicious according on how they behave. After passing a double TCP filter, if a client still launches an application layer attack, it will be detected from their I.P. addresses. BBMRD inspects these requested packets and identifies malicious and legitimate requests based on their behavior. Afterward, it forwards these requests for further measures. Binary classification is referred to those classification problems which have two class labels. For example, whether any email is spam or not. The normal state class is labeled 0, and the class with the abnormal state is assigned labeled 1. This research has employed binary classification to identify malicious and legitimate traffic. BBMRD uses different machine learning classifiers that help to identify legitimate and malicious requests based on the behavior of the packets, as shown in Figure 3. If the request is founded on highly malicious behavior, it will be declared malicious and transferred to the blacklist database. At the same time, the highly legitimate behavior request will be allowed to transfer to the request scheduler for further proceedings. Resource request pool is designed to deal with requests from the client's end and is sensitive to the Slow HTTP- post kind of DDoS attack. To avoid these issues, BBMRD has a controller who oversees this resource requests pool.

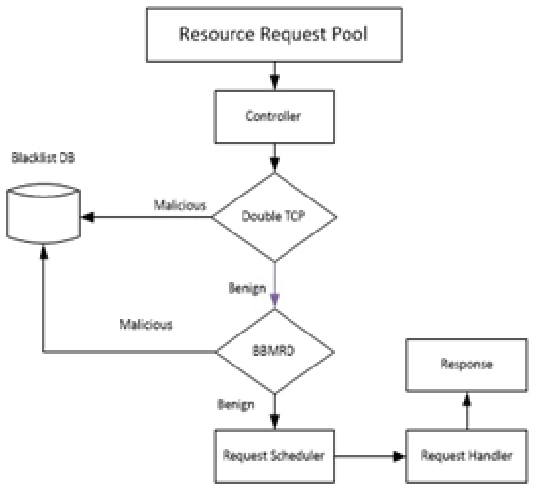

Figure 3: Behavior Based Malicious Request Detection Mechanism

A user must transmit one byte of data every two minutes to maintain the established communication. For system support, twenty seconds after the arrival of each request, the administrator monitors whether all data from the request is available. If no more data is transferred, the connection will be terminated. In this process, the blacklist database is designed for acknowledged malicious requests. At the same time, the whitelist database is designed for known benign requests. If any request finds malicious, it will be discarded. In comparison, a safe request will be forwarded for execution based on their behavior. BBMRD uses different machine learning classifiers to observe and classify the behavior of incoming traffic and identify the packet as being malicious. After that, it prevents any malicious packet from affecting the system and transfers such information into the blacklist database. Subsequently, benign packets are sent to the scheduler for further execution. Request Scheduler transfers these packets to Request-Handler for processing the packet and receiving reply to acknowledgment for the user.

5. IMPLEMENTATION

We split our proposed method into the following two phases in this research. (1) Double TCP Handshake, (2) Ensembling. Details of these phases are provided in the following paragraphs.

5.1. Double TCP Handshake

In In this phase, we used python libraries to train and validate our model. Both sides of the TCP handshake mechanism are observed, i.e., server and user end. To establish a connection between the client and the server, the Double TCP method requires a minimum of 5 packets. The server leaves the initial three-way handshake unfinished, and the subsequent three-way handshake replaces the initial attempt at connection with a new identity field and the same TCP source port number and source IP address. If found related, the proposed model allows the client to connect to the server for secure communication. otherwise, it marks it as malicious and drops. By preventing TCP SYN Flood attacks, this method makes sure that authorized users may access the server uninterrupted.

5.2. Ensembling

A DDoS evaluation dataset from the Canadian Institute for Cybersecurity (CIC) was used in the study [25]. With 99.7% of the 2.1 million observations classified as harmful and only 0.03 percent as benign, the dataset is noticeably biased towards malicious traffic. It has over 80 characteristics. This ratio depicts a real-world scenario in which a large amount of malicious traffic overtakes targets during DDoS attacks. CIC created an on-premises testbed to generate the dataset, which focuses on DDoS Lightweight Directory Access (LDAP) attacks. This ensures a controlled environment for simulated attacks. We had difficulties working with the CIC data because of its large size. Instead of using Pandas, we decided to use the Dask library to get around this limitation. Python libraries such as NumPy [24], Pandas, and sci-kit can be used for parallel processing across multiple CPU cores with the help of an open-source library called Dask. By dividing the workload into smaller parts, processing them concurrently on various CPU cores, and then aggregating the outcomes, it manages the massive dataset effectively. After processing the data, we used a variety of classification models, such as XGBoost, Random Forest, Random Forests with Bootstrap Class Weighting, Random Forest with Data Resampling (imbalanced-learn), and Weighted Logistic Regression. Even though each model worked well on its own, more predictive power was was seeking. As a result, we used the Ensembling technique, which makes use of the outputs of models that have already been created in order to improve overall predictive capabilities and produce the intended outcomes.

We used the scikit-learn library with default settings for logistic regression, which included a regularization parameter C. Additionally, we employed weighted logistic regression, in which the classes were given weights determined by how frequently they appeared in the training set. We utilized scikit-learn's "balanced" option, which automatically modifies the weights inversely proportional to the input data's class frequencies. We utilized the scikit-learn library for random forest, setting the hyperparameters n_estimators=100, max_depth=10, and random_state=42. Additionally, to address the class imbalance in the dataset, we used bootstrap class weighting. The XGBoost library was utilized, along with generic parameters like eval_metric=auc, 7 objective=binary:logistic, and booster=gbtree. We establish the subsequent booster parameters: subsample=0.8, max_depth=6, and eta=0.1. Additionally, we set task parameters like nthread=4 and num_class=1.

6. EVALUATION AND RESULTS

We use three ensemble techniques to evaluate our implementation: Hard voting, Soft-voting, and Stacked. The Hard Voting ensemble technique [26] is employed when each model makes a binary classification for the data it is fed. The ensemble model tally is up the votes to output its result. By adopting this approach, we have achieved a high accuracy rate, as seen in Figure 4. Here is how the Hard Voting Ensemble performed:

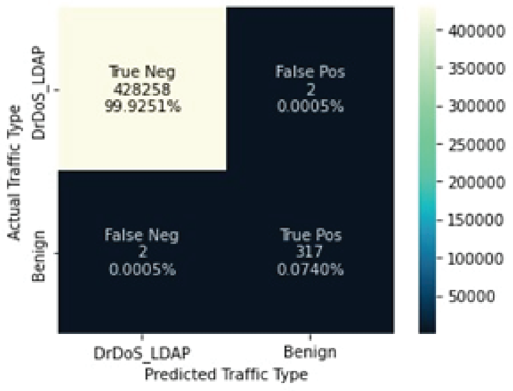

Figure 4: Hard voting ensemble confusion matrix

employs when each model outputs a probability for the malicious/ benign classification. The ensemble model then aggregates those probabilities and makes its final prediction based on which is high, as shown in Figure 5. Here is how the Soft Voting Ensemble performed.

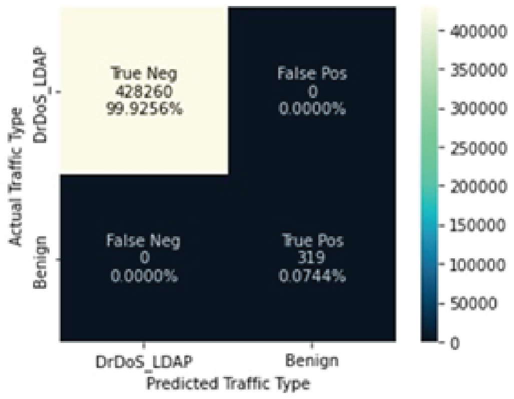

Figure 5: Soft voting ensemble confusion matrix

The Stacked ensemble [27] technique takes the wisdom of all the models in its repertoire and outputs a final prediction. It is best to give a stacked classifier model that covers each other's weaknesses instead of working against one another for the best results. The results shown in Figure 6 are well worth it.

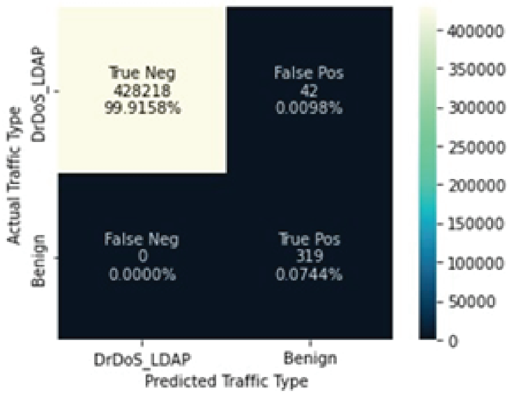

Figure 6: Stacked ensemble confusion matrix

By performing the machine learning techniques on the dataset, we calculated the F2, Recall, and Precision. Numbers show that individual classifiers do not produce the required results, as shown in Table 1.

Table 1: Individual classifier and their results

Classifier	F2	Recall	Precision
Baseline Logistic Regression	0.921	0.921	0.918
Weighted Logistic Regression	0.954	0.990	0.831
Baseline Random Forest	0.990	0.990	0.990
RFC with Bootstrap Class Weighting	0.984	0.984	0.987
Baseline XGBoost	0.986	0.986	0.946

Using the ensembling techniques resulted in better F2, Recall, and Precision, as seen in Table 2. In the evaluation section, these are discussed, which shows our final results.

Table 2: Ensembling classifier results

Classifier	F2	Recall	Precision
Hard Voting Ensemble	0.993	0.993	0.918
Soft Voting Ensemble	0.974	1.0	0.883
Stacked Ensemble	0.999	0.999	0.990

7. DISCUSSION

Cloud computing is becoming more and more vulnerable to cyberattacks, especially distributed denial-of-service (DDoS) attacks that block data transfer and services. Current methods of identifying and addressing DDoS attacks are inadequate for completely securing cloud systems against constantly changing attack techniques. In response, our study highlights improvements at the application layer and the TCP handshake protocol and suggests a technique for early DDoS detection and mitigation. Specifically designed to target DDoS attacks at the application layer, our proposed strategy uses ensemble-based machine learning approaches to categories incoming data as either legitimate or malicious. We also adopted the idea of a double TCP connection as a defense against DDoS attacks. The experimental findings demonstrate the outstanding 99.9% F-score of the stacked voting method in detecting DDoS Attacks. This study presents a useful and effective framework for early detection and mitigation of DDoS assaults in cloud computing, which helps to address the serious issue of these attacks. Our suggested approach, which makes use of machine learning and ensemble approaches, has the potential to greatly improve cybersecurity in cloud computing systems. Furthermore, the application of the double TCP connection concept strengthens the overall resilience of our architecture by adding an additional line of defense against DDoS attacks.

By putting forth a practical defense against DDoS attacks, our study makes a significant addition to cloud computing cybersecurity. The framework has proven to have strong detection accuracy and mitigation capabilities, which highlight its potential to strengthen cloud system security and resilience against DDoS attacks.

8. CONCLUSION AND FUTURE WORK

The proposed framework is an efficient and practical model that gives a significantly better outcome with better execution and higher proficiency when contrasted with the other existing and proposed frameworks with the least conceivable computational overhead. As a result of employing the Double TCP connection method, we have removed the drawback of the currently existing SYN-Caches protocol implemented in most Linux-based systems. Also, we have achieved a high detection rate of malicious requests through machine learning-based classification models. Because of the proposed method, the system can learn different types of attacks and the behavior of requests without causing damage to the system. Moreover, it ensures service availability to legitimate clients without any disturbance by rejecting malicious traffic through the blacklist database. Additionally, BBMRD has an impressive, limited number of calculations as it utilizes a whitelist database to ignore overhead in unnecessary computations for verified legitimate packets. As with all research work, there are always areas for improvement and further iteration. This research has looked at only one type of DDoS attack. Various attacks (and CIC has many datasets) have different traffic characteristics. Adding more types of malicious data and evaluating performance would benefit the model more robust to real-world scenarios. Another area of improvement would be to reduce code runtime. We would be interested in exploring the possibility of streamlining this with additional resources, intelligent code manipulation, or dimensionality reduction in the dataset.

REFERENCES

[1] R. K. Yadav et al., "Prevention Of DOS and DDoS Attack Using Count Based Filtering Method In Cloud Computing," *International Journal of Engineering Research and Technology (IJERT),* 2(6), pp.505-510, June 2013.

[2] A. S. George and S. Sagayarajan, "Securing Cloud Application Infrastructure: Understanding the Penetration Testing Challenges of IaaS, PaaS, and SaaS Environments," *Partners Universal International Research Journal,* 2(1), pp. 24-34, 2023.

[3] A. Cetinkaya et al., "An overview on denial-of-service attacks in control systems: Attack models and security analyses," *Entropy,* 21(2), pp. 210, 2019.

[4] Top Threats Working Group, "The Notorious Nine: Cloud Computing Top Threats in 2013," *Cloud Security Alliance*, February 2013.

[5] T. Lohman, "DDoS is Cloud's security Achilles heel", *Computerworld.com,* 16 September 2011.

[6] M. Prince, "The DDoS That Almost Broke the Internet," *Cloudflare,* 17 March 2013.

[7] "The Web Performance & Security Company," Cloudflare. *[Online]*. Available: https://www.Cloudflare.com/hp/. [Accessed: 11-Sep-2022].

[8] C. S. Shieh et al. “Detection of unknown ddos attacks with deep learning and gaussian mixture model,” *Applied Sciences,* 11*(*11), 5213, 2021.

[9] B. Jang et al. “Bi-LSTM model to increase accuracy in text classification: Combining Word2vec CNN and attention mechanism,” *Applied Sciences,* 10(17), pp. 5841, 2020.

[10] C. L. Schuba et al., “Analysis of a denial of service attack on TCP. In Proceedings,” *1997 IEEE Symposium on Security and Privacy (Cat. No. 97CB36097),* pp. 208-223, IEEE, May, 1997.

[11] M. M. Alani, “OSI model, Guide to OSI and TCP/IP Models,” *Springer, Cham,* pp. 5-17, 2014.

[12] C. Espositheo et al., “Interconnecting Federated Clouds by using Publish-Subscribe Service,” *Cluster Compu., Springer* 2013.

[13] K. Modi et al., “Detection and Prevention of DDoS Attacks on the Cloud using Double-TCP Mechanism and HMM-based architecture,” *International Journal of Cloud Computing and Services Science,* 3(2), pp.113, April 2014.

[14] J. Shahid et al., “A Comparative Study of Web Application Security Parameters: Current Trends and Future Directions,” *Applied Sciences,* 12(8), pp. 4077, 2022.

[15] S. Ali et al., “Comparative Evaluation of AI-Based Techniques for Zero-Day Attacks Detection,” *Electronics,* 11(23), pp. 3934, 2022.

[16] A. Khajuria and R. Srivastava, “Analysis of the DDoS Defense Strategies in Cloud Computing,” *International Journal of Enhanced Research in Management and Computer Appliattackss*, 2(2), Feb 2013.

[17] B. Singh et al., “Threshold-based Approach to Detect DDoS Attacks in Cloud,” *International Journal of Innovative Research in Information Security (IJIRIS)*, 3(2): March 2015.

[18] D. S. SaravanaKumar et al., “Detecting and Preventing DDoS Attacks in Cloud,” I*nternational Journal of Innovative Research in Computer and Communication Engineering*, 3(3), March 2015.

[19] Patel Ankita and Fenil Khatiwala,” Survey on DDoS Attack Detection and Prevention in Cloud”, International Journal of Engineering Technology, Management, and Applied Sciences, 3(2): February 2015.

[20] M. A. Aladaileh et al., “Detection techniques of distributed denial of service attacks on software-defined networking controller–a review,” *IEEE Access,* 8, pp. 143985-143995, 2020,

[21] A. A. Alashhab et al., “A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks,” *Symmetry,* 14(8), pp. 1563, 2022.

[22] G. Preetha et al., “Autonomous Agent for DDoS Attack Detection and Defense in an Experimental Testbed,” *International Journal of Fuzzy Systems,* 16(4), December 2014.

[23] S. Subapriya and M. N. Radhika, “DNIDPS: Distributed Network Intrusion Detection and Prevention System,” *IJISET- International Journal of Innovative Science, Engineering and Technology,* 1(7), pp.56-67, September 2014.

[24] J. Rameshbabu et al., “Prevention of DDoS Attack in Cloud using NEIF Techniques,” *International Journal of Scientific and Research Publications*, 4(4), pp.1-5, April 2014.

[25] “DDoS Evaluation Dataset,” *[Online]*, Availabale: https://www.unb.ca/cic/datasets/index.html. [Accessed: 11-Sep-2022].

[26] J. A. Morgan-Benita et al., “Hard Voting Ensemble Approach for the Detection of Type 2 Diabetes in Mexican Population with Non-Glucose Related Features,” *Healthcare,* 10(8), Multidisciplinary Digital Publishing Institute, 2022.

[27] S. Kumari et al., “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” *International Journal of Cognitive Computing in Engineering,* 2, pp. 40-46, 2021.