ABUAD Journal of Engineering Research and Development (AJERD)
ISSN (online): 2645-2685; ISSN (print): 2756-6811

Volume 8, Issue 2, 14-24

Principal Component Analysis-Multilinear Perceptron-based model for Distributed Denial of Service Attack Mitigation

Opeyemi Oreoluwa ASAOLU[1], Oluwasanmi Segun ADANIGBO[2], Afeez Adekunle SOLADOYE[1], Nnamdi Stephen OKOMBA[1]

[1]*Department of Computer Engineering, Federal University of Oye-Ekiti, Ekiti State, Nigeria*
opeyemi.adanigbo@fuoye.edu.ng/sabdulhafeedh@gmail.com/nnamdi.okomba@fuoye.edu.ng

[2]*School of Management (Fin Tech), University of Bradford, West Yorkshire, BD7 1DP, UK*
sanmiadas@gmail.com

Corresponding Author: opeyemi.adanigbo@fuoye.edu.ng, +2347068505713
Date Submitted: 05/12/2024
Date Accepted: 03/05/2025
Date Published: 06/05/2025

Abstract: *The increasing occurrence of Distributed Denial of Service (DDoS) attacks has caused significant disruptions in global network services, overwhelming targets by flooding them with requests from various sources. This ease of execution and gaining entry to distributed systems for rent has led to increasing financial losses. This paper addresses the growing challenge of* IoT devices-targeted Distributed Denial of Service (DDoS) attacks *within 4G networks. In this study, a PCA-MLP (Principal Component Analysis-Multi-Layer Perceptron) intrusion detection model combined with a packet-filtering firewall for enhanced prevention is presented. The firewall, utilizing IPtables, selectively permits traffic from trusted sources, successfully blocking nearly 70% of DDoS threats. The PCA-MLP model proposed in this study demonstrated high performance, accurately identifying different types of DDoS attacks with an overall accuracy of 95.35%.*

Keywords: *DDoS Attacks, Firewall, Intrusion Detection, IoT Devices, PCA-MLP, 4G Networks*

1. INTRODUCTION

The swift expansion of the Internet of Things (IoT) has brought numerous advantages and conveniences. However, the extensive network of interconnected devices also creates substantial security risks. IoT equipment, ranging from personal electronics to industrial systems, are particularly vulnerable to cyberattacks [1]. Among such threats, Distributed Denial of Service (DDoS) attacks present a major challenge, especially for IoT connectivity on widely used 4G networks [2]. Such attacks have become a serious concern for businesses that rely on public networks for their technological integration (Manita and Suleiman, 2024). To address these challenges, an effective security system is necessary to identify and counteract DDoS attacks targeting IoT devices utilizing 4G, ensuring both reliability and scalability. These attacks can severely impair system and service performance [3].

The widespread adoption of IoT devices has made them increasingly susceptible to DDoS and DoS attacks, largely due to insufficient security protocols. The vast number of connected devices, combined with their limited memory and processing capabilities, heightens the vulnerability of 4G networks. Furthermore, the unique features of IoT devices, including low computational capacity and diverse communication protocols, add complexity to security management. To mitigate these risks, a robust security framework is required, incorporating firewall defenses, packet filtering, and multi-layered protection at both the device and network levels [4].

Existing countermeasures such as traditional firewalls and machine learning (ML)-based intrusion detection systems have made strides in mitigating these threats. However, many of these approaches suffer from limitations including low detection accuracy, lack of adaptability to new attack patterns, and insufficient real-time processing capabilities. To address these challenges, this paper proposes a hybrid model that combines Principal Component Analysis (PCA) for dimensionality reduction with a Multi-Layer Perceptron (MLP) for accurate classification. The model is integrated with a real-time packet-filtering firewall (IPtables) to further strengthen its defence capabilities. This approach achieved an overall detection accuracy of 95.35%, demonstrating its effectiveness in identifying and mitigating various types of DDoS attacks. This study utilizes network-level strategies such as traffic filtering, rate limiting, and IP verification to combat malicious activity. The remainder of this paper is organized as follows: Section 2 presents related works and a comprehensive literature review, highlighting current research gaps. Section 3 details our methodology, including the mathematical model, dataset description, and experimental setup. Section 4 presents the results and discussion, including comparative analysis with existing approaches. Section 5 acknowledges the limitations of the current study, while Section

6 discusses the integration and adaptability of the proposed solution. Finally, Section 7 provides conclusions and directions for future research.

2. RELATED WORKS

This study introduces a method that filters incoming packets employing a firewall to protect in opposition to DDoS attacks. Automated firewall configurations were carried out with the Linux command-line tool iptables to block suspicious traffic effectively. The combination of Wireshark, a network traffic analysis tool, and an iptables-based firewall system demonstrated strong capabilities for monitoring and mitigating various types of DDoS attacks. By dynamically addressing potential threats identified by Wireshark, this approach adds an additional layer of security to the server. The types of network attacks analysed in this study are detailed in the subsequent sections.

i. Denial of Service (DOS): DoS attacks represent a major risk to subscriber networks, targeting their operations with the intent to cause temporary or permanent disruptions. These attacks are typically carried out by individuals or organizations aiming to disable access to specific websites or services. High-priority targets often include essential online hosting platforms, including those handling credit card-based transaction processing and banking services [5].
ii. Distributed Denial of Service: A Distributed Denial of Service (DDoS) attack seeks to render a service inaccessible by inundating the server with a large volume of harmful traffic [6].
iii. Distributed Reflection Denial of Service-Network Time Protocol (DrDoS_NTP): DrDoS_NTP is a type of attack that takes advantage of weaknesses in the NTP system. In this type of attack, perpetrators send spoofed requests to NTP servers, prompting them to overwhelm the target server with amplified traffic. This amplification happens because the NTP servers respond with data significantly larger than the initial requests [7].
iv. Trivial File Transfer Protocol (TFTP): is a streamlined protocol created for transferring files, primarily within Local Area Network environments [8]. It uses UDP and does not include authentication features, making it vulnerable to exploitation in cyberattacks like DrDoS attacks. In these scenarios, attackers exploit TFTP servers by sending minimal requests that trigger disproportionately large responses, amplifying the traffic.
v. Benign traffic: Benign traffic consists of legitimate and harmless data packets that do not pose any threat to network integrity. It serves as a baseline to define normal network behaviour, enabling intrusion detection systems to identify anomalies effectively [9].
vi. Synchronize (SYN): SYN is the initial step in the TCP three-way protocol initialization process, in which a client sends a synchronize (SYN) packet to server to start a connection. In network attacks, a SYN flood occurs when an attacker bombards a server with excessive SYN packets, depleting its resources and blocking legitimate connections [10].

2.1 Literature Review

A study by Nanda *et al*. [11] proposed the use of machine learning algorithms trained on historical network attack data to predict and block possible harmful connections as well as their targets. The researchers employed four commonly recognized machine learning techniques— Bayesian Network, Naive Bayes, C4.5, and Decision Table algorithms to forecast likely attack targets. Their experiments unveiled that Bayesian Networks attained an average predictive accuracy of 91.68%. Another study by Kodati *et al.* [12] focused on detecting Detecting and mitigating distributed denial of service (DDoS) attacks in software-defined networks (SDNs) through machine learning techniques. The models evaluated included K-Nearest Neighbour (85.4% accuracy), Naive Bayes (84.2%), Support Vector Machines (86.4%), and Decision Tree (88.1%). Similarly, Tzagkarakis *et al.* [13] developed a framework for identifying compromised IoT devices within a botnet and detecting malicious activities at the IoT-Edge layer. The framework utilized Sparsity Representation and Reconstruction Error Threshold methods for traffic classification, leveraging only threshold error analysis since the machine learning models were trained on benign traffic data from NB-IoT.

Liu *et al.* [14] introduced a two-tier DDoS detection strategy tailored for SDN environments. The first tier employed an entropy-based technique to identify potentially suspicious components and ports, while the second tier applied a Convolutional Neural Network (CNN) model for more precise detection, distinguishing between legitimate and malicious communications. Another framework, Salim *et al.* [15] utilized the Long Short-Term Memory (LSTM) algorithm and deep learning techniques to identify firmware vulnerabilities in IoT devices. This system analysed and disabled compromised devices, continuously monitored them, and re-enabled them once they were secure. Hussain *et al.* [16] proposed a method for identifying various DDoS attack types, such as SMS flooding, signalling, and silent call attacks, in 4G LTE-A networks. This approach combined a Convolutional Neural Network for pre-processing traffic in the network core with a ResNet-50 model to classify traffic and detect attack types. The deep learning model was trained on a dataset from Telecom India. Sahi *et al*. [17] outlined a classification framework for identifying and mitigate DDoS TCP flood threats within public cloud environments. The system categorized packets and made mitigation decisions based on the classifications to secure stored records. Additionally, Ozer *et al.* [18] suggested a method using deep packet analysis for rapid DDoS detection. This involved capturing network traffic packets, filtering them at a pre-set threshold, logging them in a database, and comparing average values to known DDoS patterns. Osanaiye *et al.* [19] Presented a DDoS detection system employing a multi-filter feature selection approach based on ensemble methods. By combining outputs from four different filter methods, the system optimized feature selection. Using the NSL-KDD data set and a classifier based on a

decision tree, the system demonstrated efficiency in feature selection but was limited by processing delays.

Almadhor *et al.* [20] focused on developing an anomaly detection system utilizing machine learning to mitigate DDoS attacks in IoT networks. By employing a diverse dataset encompassing various IoT device types and network configurations, the research evaluated multiple machine learning algorithms for DDoS detection. The system demonstrated high accuracy in identifying abnormal traffic patterns associated with DDoS attacks. While the study achieved commendable accuracy, it did not integrate real-time response mechanisms, such as firewall interventions, to actively mitigate detected threats. Additionally, the adaptability of the model to evolving attack vectors remains unaddressed. Also, Alahmadi *et al.* [21] employed an improved feature selection technique to identify the most relevant features for DDoS detection in Software-Defined IoT (SDIoT) environments. By reducing the feature space, the study aimed to improve detection rates and reduce training times of machine learning classifiers. Although effective in feature selection, the study focused primarily on SDIoT environments, which may not directly translate to traditional IoT over 4G networks.

Furthermore, the integration of the detection model with real-time mitigation strategies, such as firewall implementations, was not explored. In Saravanan *et al.* [22], the study introduced a federated learning approach combined with Explainable Artificial Intelligence (XAI) to detect DDoS attacks in heterogeneous IoT environments. The model aimed to enhance detection accuracy while preserving data privacy across distributed IoT devices. While the federated learning approach addresses data privacy concerns, it may introduce complexities in model synchronization and increased communication overhead. Additionally, the study does not incorporate real-time mitigation measures post-detection. The PCA-MLP hybrid model proposed in this study not only enhances detection accuracy but also incorporates real-time packet-filtering firewalls, enabling immediate response to identified threats and improving adaptability to new attack patterns. This research extends the application of effective feature selection to IoT over 4G networks and integrates the detection model with real-time firewall responses, ensuring both efficient detection and immediate mitigation of DDoS attacks. Finally, the PCA-MLP hybrid model focuses on centralized processing to simplify implementation and integrates real-time packet-filtering mechanisms, providing both accurate detection and immediate response to DDoS threats in IoT over 4G networks.

3. METHODOLOGY

The experimental setup was configured using Linux Ubuntu Server version 10.10 with Wireshark installed, a tool for analysing network protocols. The model architectural framework of the model is illustrated in Figure 1.

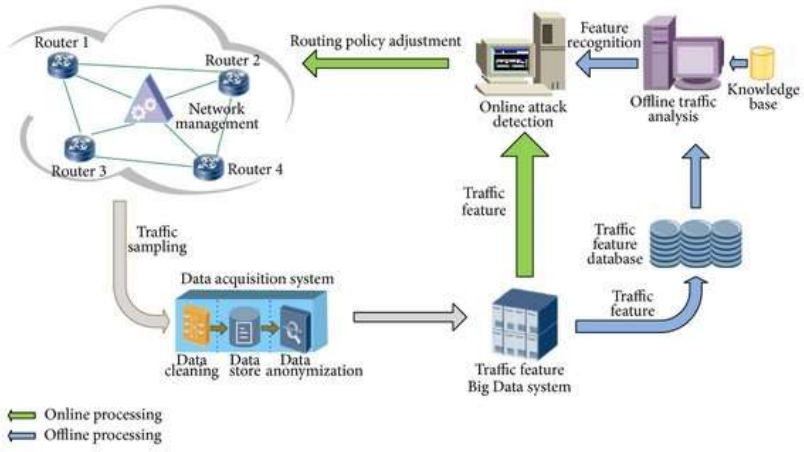

Figure 1: The architectural framework of the model.

3.1 System Design

The experimental configuration utilized a Linux Ubuntu Server version 10.10 alongside Wireshark, a tool for network protocol analysis. A system was used to serve as an interface between the Wireshark application, used for network traffic monitoring and analysis, and an iptables-based firewall system. This system processes the output from Wireshark to identify potentially malicious IP addresses. When a malicious IP is detected, the system issues a command to iptables to add a rule blocking the identified IP. Similarly, if the analysis identifies a compromised port, the system adds a rule to close that port. Conversely, if the analysis determines an IP address or port is no longer a threat, the system unblocks them accordingly.

To optimize the dataset, dimensionality reduction was performed using a filtering method based on information gain with ranker. This approach evaluates the entropy-based information gain for each attribute, ranking them in descending order of relevance. Each attribute is assigned an evaluation metric beginning with 1 (most significant) and ending at 0

(least significant), with the highest-ranked attributes forming the input subset for the next stage of dimensionality reduction. The flow of this model is illustrated in Figure 2.

3.2 Mathematical Model

The dimension of the dataset used was reduced by employing the information gain method with a ranker serving as a filtering approach. This technique ranks attribute subsets by calculating their information gain entropy in descending order. All attributes are assigned scores ranging from 1 (indicating the most relevant) to 0 (indicating the least relevant). Features with the highest scores are selected as part of the input features for the subsequent dimension reduction phase.

If m represents the number of classes, d_i denotes the number of instances belonging to class i, and D is the total number of instances in the training set, Equation 1 calculates the estimated information required to classify a given instance.

$$I(d_1, d_2, \dots d_m) = -\sum_{i=1}^{m} \frac{d_i}{D} . log_2 \left(\frac{d_i}{D}\right) \quad (1)$$

An attribute T with values $\{t_1; t_2; \dots ; t_v\}$ was used in dividing the training data into v partitions:$\{S_1,S_2,\dots,S_v\}$.where S_j is

Where m: Total number of classes, d_i: Number of instances belonging to class i, D: Total number of instances in the dataset d_i / D: Probability of an instance belonging to class i, and I(·): Information entropy, (i.e., expected amount of information needed to classify a new instance).

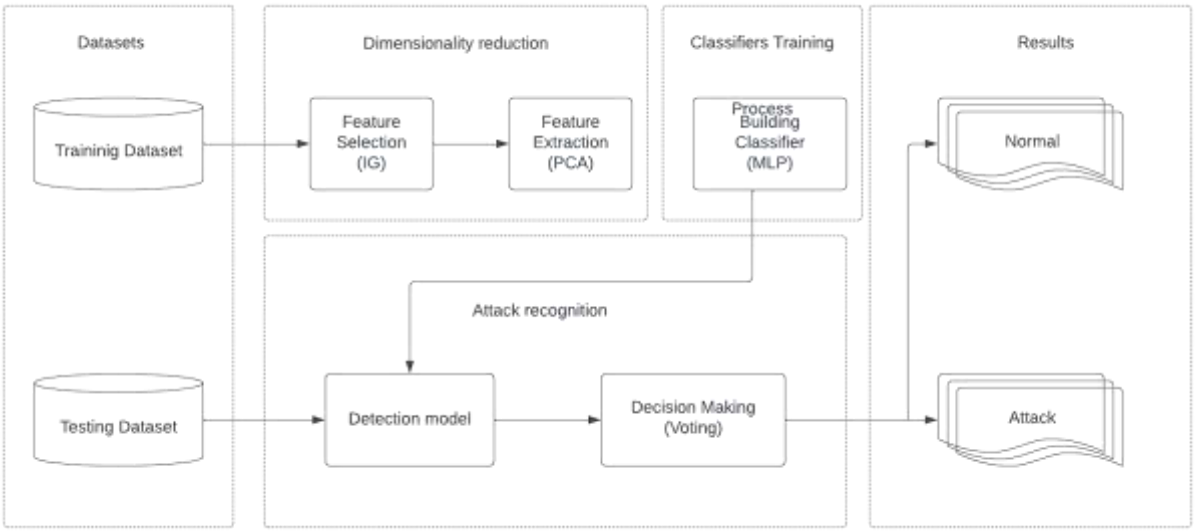

Figure 2: Model flow process

Given that $I(x_1, x_2, \dots x_m)$ represents the information required to classify a given instance with attribute values x_1 through x_m κ is the number of classes in the classification problem, π_i represents the number of instances belonging to class i in the training data, T is the total number of instances in the training set, π_i / T represents the probability of an instance belonging to class i, and $\log_2(\pi_i / T)$ is the logarithm (base 2) of this probability T.

Furthermore, if S_j includes d_{ij} instances of class I, the entropy of the feature T is calculated as:

$$E(T) = \sum_{j=1}^{v} \frac{d_{1j}+\dots+d_{mj}}{D} \times I\left(d_{1j} \dots d_{mj}\right) \quad (2)$$

Where T: Attribute under consideration, v: Number of distinct values of attribute T, S_j: Subset of data where T takes the j^{th} value, d_{ij}: Number of instances of class i in subset S_j, D: Total number of instances, and $I(d_{1j}, \dots, d_{mj})$: Entropy of the j^{th} subset. Equation 3 presents the computation needed to derive the information gain (IG) for an attribute T:

$$I(Gain(T) = d_1, \dots, d_m) - E(T) \quad (3)$$

Where Gain(T): Information gain of attribute T, $I(d_1, \dots, d_m)$: Entropy before splitting, and E(T): Entropy after splitting on attribute T.

PCA was used in lowering the dimension of the data to enhance the detection of DDoS attacks. The process can be described mathematically as follows.

If {x(t)} for t = 1…n represents a randomized dataset consisting of matching instances and attributes with Mean normalized to zero. The covariance matrix of x(t) is expressed as presented in Equation 4:

$$R = \frac{1}{n-1} \sum_{i=1}^{n} [x(t) \times (t)^T] \quad (4)$$

Where R depicts the Covariance matrix, x(t) is the Data sample (feature vector) at time t, assumed zero-mean, n is number of samples, $x(t)^t$ is the transpose of x(t), and × is the Outer product (vector multiplication, not element-wise).

In Principal Component Analysis (PCA), the linear mapping from x(t) to y(t) can be calculated as:

$$y(t) = M^T \times (t) \quad (5)$$

Where y(t) is Transformed feature vector (principal components), x(t) represents the Original input feature vector, M is the matrix of eigenvectors (principal directions), and M^t is the Transpose of M used for projection.

In Principal Component Analysis (PCA), M represents an $n \times n$ orthogonal matrix where the i^{th} column corresponds to the i^{th} eigenvector of the sample covariance matrix R. PCA solves the eigenvalue problem to determine these eigenvectors and eigenvalues. The eigenvalue problem is expressed as:

$$\lambda_i q_i = R q_i \quad (6)$$

Where λ_i represents the i^{th} eigenvalue of covariance matrix R, q_i is the i^{th} eigenvector of R, and R is the Covariance matrix. Given that λ_i represents an eigenvalue of the covariance matrix R, whereas the associated eigenvector is represented by q_i. According to Equation 5, the principal component is obtained by:

$$y_i(t) = q_i^T x(t), i = 1, \dots, n \quad (7)$$

Where $y_i(t)$ signifies the i^{th} principal component at time t, q_i: i^{th} eigenvector, and x(t): Original input feature vector. The principal directions (the top k eigenvectors) are obtained by sorting the eigenvalues λ_i in descending order. These eigenvectors are used for feature extraction, capturing the most significant variance in the data. Equation 8 defines the calculation for projecting a new sample x(t) onto the primary subspace as:

$$\varkappa(t) = \sum_{i=1}^{k} b_i^T x(t) b_i \quad (8)$$

Given that B = (b_i : $b_i = q_i$; i = 1; ... ; k}, χ(t) is the Projection of x(t) onto the k-dimensional principal subspace, b_i is the i^{th} selected eigenvector (same as q_i), b_i^t x(t) is the scalar projection of x(t) onto b_i , and k is the number of top principal components. Equation 9 demonstrates the mapping error of x(t) by determining the distance d between x(t) and, $\varkappa(t)$:

$$e_t = d\big(x(t), \varkappa(t)\big) \quad (9)$$

Where e_t= Projection or reconstruction error at time t, x(t)= Original data sample, χ(t) = Projected sample, and d(x(t), χ(t)) = Distance between original and projected data (e.g., Euclidean distance). The algorithm for the Principal Component Analysis (PCA) procedure is summarized in Table 1.

Table 1: Feature selection and dimensionality reduction

Table 1: Feature Selection and Dimensionality Reduction
Input:
Dataset X, where X consists of n samples, each containing T features.
Subroutine: COMPUTE_IG (X)
Determine the Entropy:
Determine the estimated amount of information necessary to classify a specific instance.
For Each Attribute T_i:
Compute the entropy of T_i.
Compute the information gain of T_i.
Select Top k Attributes:
Select the k attributes with the highest significant information gain values.
Let Y represent these top k attributes.
Return Y:
The subset of selected features.
Subroutine: PERFORM_PCA(Y)
Calculate the Covariance Matrix:
Calculate the matrix for obtaining the covariance of the feature subset Y.
Compute Eigenvalues and Eigenvectors:
Extract eigenvectors alongside eigenvalues (λ_1,…,λ_i) from the covariation matrix.
Select Principal Components:
Identify the k eigenvectors corresponding to the largest eigenvalues.
Let Z represent these k-dimensional principal components.

Output Z:
The newly formed k-dimensional feature space.

Output:
The transformed dataset Z, reduced to k-dimensions for further processing.

3.3 Dataset

The PCA-MLP model, combining Principal Component Analysis for dimensionality reduction and Multilayer Perceptron for learning and classification, was employed in detecting DoS attacks. The CIC-DDOS2019 dataset, provided by the Canadian Institute for Cybersecurity was downloaded online at: https://www.unb.ca/cic/datasets/ddos-2019.html, included the CICDDoS2019 statistics, encompassing various DoS and DDoS attacks along with multistage attacks. The dataset used, Pre-processed using the CICFlowMeter, contained 88 network traffic features recorded in CSV format. The dataset used contained 73,498 samples, 19,448 samples of DrDoS_NTP attacks, 24,308 of TFTP, 9,934 of benign and 19,808 samples of syn. The CIC-DDOS2019 dataset was split into two distinct subsets. Approximately 20% of the dataset was allocated for testing, while the remaining 80% was designated for training purposes. Python's scikit-learn library provided the foundational tools for implementing the PCA-MLP hybrid learning model. The training process carried out using Google Colab, a cloud-based environment known for its computational resources and compatibility with the required machine learning libraries.

3.4 Training and Evaluation

The training and evaluation phase of the PCA-MLP hybrid model involved fine-tuning essential hyper parameters to achieve optimal performance in detecting Denial-of-Service (DoS) attacks. The model was constructed with a hidden layer architecture of (10, 10), meaning two concealed layers, each comprising 10 neurons, utilizing the Rectified Linear Unit (ReLU) activation function to incorporate non-linearity to the network, enhancing its ability to learn intricate patterns from the data. The 'adam' solver was chosen to optimize the neural network's weights during training. The model was trained for a maximum of 1000 iterations to facilitate convergence. Furthermore, Principal Component Analysis was employed to decrease the input data's dimensionality, allowing the model to concentrate on the most significant features.

3.5 Performance Metrics

The precision shows how many positive predictions are predicted correctly.

$$Precision(pre) = \frac{TP}{TP+FP} \tag{10}$$

The recall value is determined using Equation 11, which indicates the proportion of true positives correctly identified.

$$Recall \vee DetectionRate(DR) = \frac{TP}{TP+FN} \tag{11}$$

Equation 12 calculates the accuracy, revealing the model's accuracy in making correct predictions.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{12}$$

Equation 13 shows the F1 score used to assess the effectiveness of a classification model, especially in cases where there is an imbalance between the classes. It considers both precision and recall, providing a balanced measure of the model's accuracy.

$$F1Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{13}$$

Where TP = True Positive, FP = False Positive, TN = True Negative, FN = False Negative.

4. RESULTS AND DISCUSSION

Wireshark was used continuously to log data transmitted through the adapter connected to the server. When abnormal or suspicious activity was detected, the system generated a prompt for the user to either close the affected port or block the associated IP address. This anomalous data was then processed by an independent application that created IPtables rules for the firewall. The scanning results from Wireshark were saved to a file, which the program used to generate these rules. A Python GUI application, built with Tkinter, provided a user-friendly interface for the server administrator to process Wireshark's output and apply the necessary firewall rules. The model flowchart is presented in Figure 3.

To evaluate the system's effectiveness against DOS attacks, simulated attacks were conducted using tools such as Hping3, Nmap, and Synflood. The tests compared performance on a server equipped with the program against one without it. The simulation setup included a laptop (acting as the attacker) and a computer (the host/server), connected via a home Wi-Fi modem with internet access provided by a 4G router. The target system (host) was configured with an IPtables firewall, Wireshark for monitoring, and applications to detect network anomalies and update firewall rules in real-time. The laptop, acting as the attacker, utilized tools like Nmap for network mapping, Hping3 for crafting and sending custom

packets, and Synflood for executing attack simulations aimed at overwhelming network traffic. During the Synflood attacks, network congestion was introduced, creating anomalies detected by Wireshark. An application processed Wireshark's output and updated IPtables with new firewall rules to strengthen security. The results showed that the number of firewall rules had minimal impact on UDP traffic. Throughput rates were influenced by packet size: larger packets resulted in higher throughput, while smaller packets yielded lower rates. The maximum throughput was observed with a packet size of 1 kilobyte, but performance declined when the size increased to 1.5 kilobytes.

This flowchart displayed in Figure 3 provides a concise overview of the DDoS mitigation workflow using the CIC-DDoS2019 dataset. It starts with data pre-processing, where the dataset is split into training and testing subsets. Each subset undergoes normalization to ensure consistent feature scaling. The training set is labelled as 'Normal' or 'Attack' and used to train the model. The testing set is passed through a firewall model, which applies the trained classifier to detect threats. The model's performance is then evaluated using precision, recall, F1 score, and accuracy metrics. This flow effectively demonstrates the end-to-end pipeline from data acquisition to performance evaluation and firewall deployment.

4.1 Latency

The results show that increasing the number of filtering rules on firewalls leads to a slight decrease in the transaction rate, indicating degraded performance as the rules are increased.

4.2 Firewall Performance

Repetitive simulation processes over a set period of time showed that the collaborative algorithm between Wireshark and IPtables worked effectively. Approximately 70% of IP threats, especially those using Nmap for open port scanning and Hping3 for attacks, were successfully blocked. Figure 3 highlights the trade-off between processing efficiency and the complexity introduced by filtering rules and data payload sizes. In Figures 4 and 5, various configurations of request (Req) and response (Res) sizes, were used both for low performance (throughput) system and high performance (throughput) system. Generally, as the number of filtering rules increases (from 0 to 1000), the transaction rate per second decreases for most configurations. This indicates that additional filtering rules impose a computational overhead, slowing down transaction processing. However, configurations with larger payloads (such as Req=512, Res=1024) show significant performance degradation. The configuration with the smallest payload size (Req=1, Res=1) consistently achieves the highest transaction rates across all levels of filtering rules while the configuration with the largest payload size (Req=512, Res=1024) has the lowest transaction rates, especially as the number of rules increases. Figure 6 presents the general model flowchart. A summary of performance metrics used on different DoS attack classes in this study is presented in Table 2.

Table 2: Summary of performance metrics on different DoS attack classes

Label	**Precision(%)**	**Recall(%)**	**F1 Score(%)**	Accuracy
DrDoS_NTP	92.90	78.40	85.00	92.70
TFTP	93.40	96.30	94.80	96.50
Benign	80.00	88.30	83.90	95.40
Syn	91.60	97.20	94.30	96.80

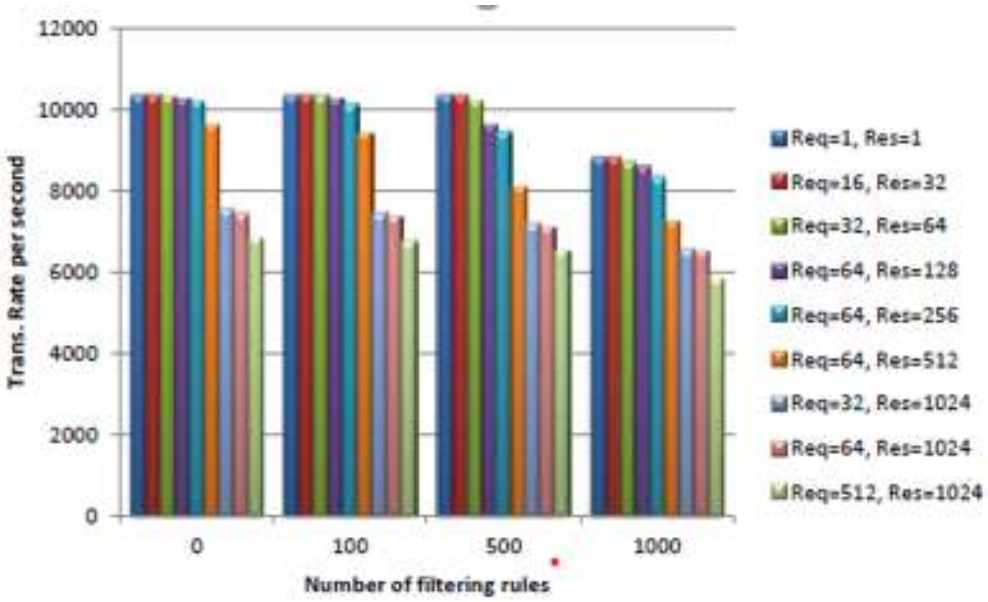

Figure 3: Transaction (Trans.) versus number of filtering rules

Figure 4: Transmission rate per sec vs. Number of filtering rules for a low throughput System

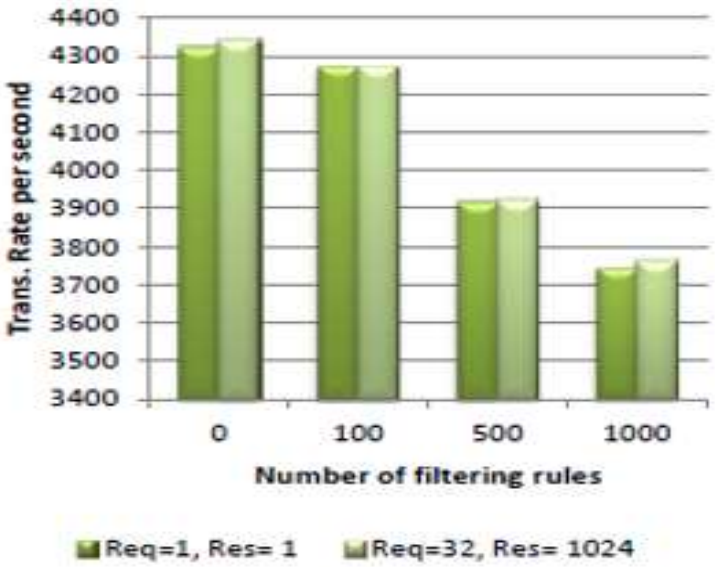

Figure 5: Transmission rate per sec vs. Number of filtering rules for a high throughput System

The confusion matrix is shown in Figure 7. It shows that class 0 made 15,250 correct predictions. class 1 made 23,419 correct predictions, class 2 made 8,776 correct predictions while class 3 made 19,251 correct predictions. The Precision-Recall graph shown in Figure 8 demonstrates the balance across precision and recall at various probability thresholds, emphasizing positive class prediction quality, making it particularly useful in scenarios with imbalanced classes.

4.3 Comparison of Proposed Model with Recent Studies

A table comparing the results obtained from this study with a number of recent studies is presented in Table 3.

4.4 Limitations of the Study

While this study demonstrates the effectiveness of a PCA-MLP model integrated with a real-time firewall for DDoS mitigation, it has certain limitations. The approach was validated in a simulated environment, and real-world deployment has not been tested. Additionally, the model's performance is evaluated on a single dataset (CIC-DDoS2019), and its generalizability to other datasets or live traffic scenarios remains to be explored.

While these simulation results are promising, real-world network environments present additional challenges not fully captured in the controlled testing environment. Factors such as variable network conditions, diverse traffic patterns, and hardware constraints may impact the model's performance in production settings. Future work will focus on deploying the system in limited real-world scenarios to validate its performance under actual network conditions and to identify potential implementation challenges that may not be apparent in simulations.

Figure 6: Flowchart of the PCA-MLP model

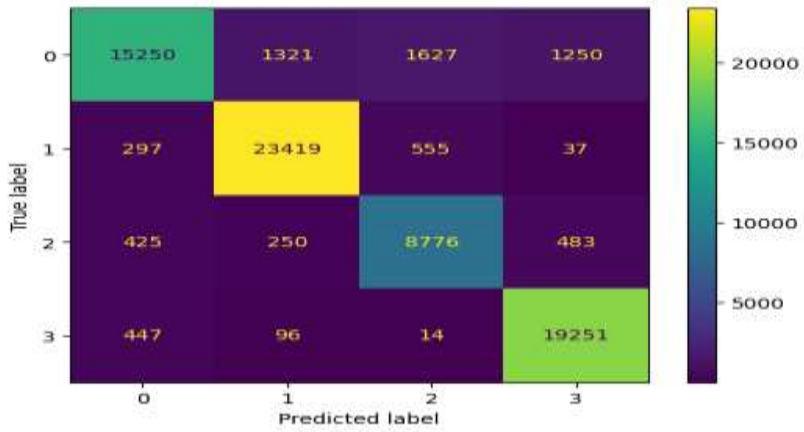

Figure 7: Confusion matrix of the model

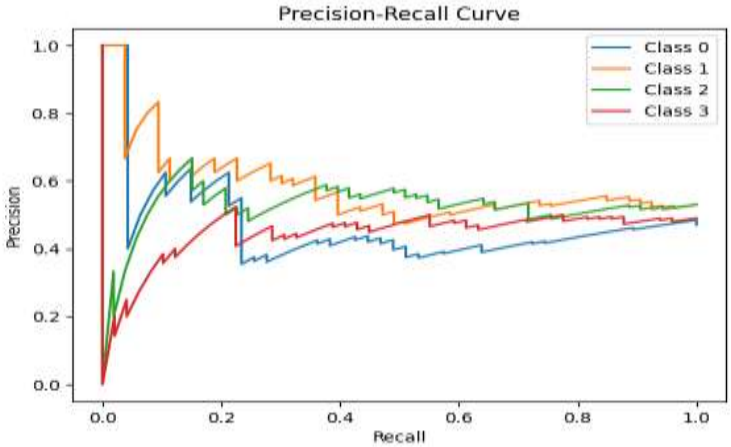

Figure 8: Precision-Recall curve of the model

Table 3: Comparison of proposed model with recent studies

Model	Dataset	Accuracy (%)	Real-Time Mitigation	Remarks
Bayesian Network [11]	SDN Attack Data	91.68	No	Good predictive accuracy, but lacks real-time mitigation
Decision Tree [12]	SDN	88.1	No	Moderate accuracy; lacks firewall integration
CNN + Entropy [14]	SDN	94.1	No	Good precision; limited to SDN
Federated XAI [22]	Heterogeneous IoT	93.7	No	Preserves data privacy but lacks deployment ease
Proposed PCA-MLP	CIC-DDoS2019	95.35	Yes	High accuracy and real-time IPtables firewall integration

4.5 Integration and Adaptability of the Proposed Solution

The proposed PCA-MLP-based detection system is designed for compatibility with existing Linux-based network infrastructures. By leveraging standard tools such as Wireshark and IPtables, it can be easily integrated into real-time monitoring setups. The system is adaptive, dynamically generating firewall rules in response to detected threats, thereby reducing reliance on static configurations. This adaptability allows for real-time updates and responsiveness to emerging attack vectors, making it suitable for evolving IoT environments.

5. CONCLUSION

This study presents an integrated PCA-MLP model with a real-time firewall mechanism for mitigating DDoS attacks on IoT devices over 4G networks. Unlike existing methods, the proposed approach combines intelligent detection with active prevention, achieving high accuracy while dynamically filtering malicious traffic. This hybrid system improves security, scalability, and responsiveness in resource-constrained environments. The work advances current research by bridging the gap between detection and real-time mitigation, offering a practical solution for modern IoT security challenges. The results demonstrated that larger packet sizes achieved higher throughput rates; however, an increase in the number of filtering rules caused a slight decline in transaction rates. In summary, the proposed packet filtering approach, incorporating a firewall and the PCA-MLP hybrid model, provides an efficient method for detecting and mitigating DoS/DDoS attacks. Future research could aim to enhance the system's long-term performance, refine detection accuracy, and minimize false positives by exploring additional optimization techniques.

REFERENCES

[1] Ali, M. S., Shah, S. A., Hussain, A., & Wadhwani, S. K. (2022). A novel hybrid approach for lightweight device fingerprinting and anomaly detection in IoT networks. *Sensors, 22*(3), 822. https://doi.org/10.3390/s22030822

[2] Kumar, S., & Kumar, R. (2016). Denial of service attacks: A comprehensive study. *Procedia Computer Science, 78*, 396–401. https://doi.org/10.1016/j.procs.2016.02.068

[3] Farhan, M. A., Khan, S. U., & Salah, K. (2018). A review on mitigation techniques for distributed denial-of-service (DDoS) attacks in IoT networks. *IEEE Communications Surveys & Tutorials, 20*(4), 2058–2085. https://doi.org/10.1109/COMST.2018.2844341

[4] Bica, I., Chifor, B.-C., Arseni, Ș.-C., & Matei, I. (2019). Multi-layer IoT security framework for ambient intelligence environments. *Sensors, 19*(18), 4038. https://doi.org/10.3390/s19184038

[5] Rao, G., & Subbarao, P. (2023). A novel approach for detection of DoS/DDoS attack in network environment using ensemble machine learning model. *International Journal on Recent and Innovation Trends in Computing and Communication, 11*, 244–253. https://doi.org/10.17762/ijritcc.v11i9.8340

[6] Tangtode, D., Sayyad, S., Gelye, O., Sawant, S., & Bombale, P. (2024). DDOS attack detection. *International Journal of Advanced Research in Science, Communication and Technology*, 248–251. https://doi.org/10.48175/IJARSCT-15547

[7] Rossow, C. (2014). Amplification hell: Revisiting network protocols for DDoS abuse. *Proceedings of the Network and Distributed System Security Symposium (NDSS).*

[8] Tandem Computers. (2006). *Securing HP NonStop servers in an open systems world: TCP/IP, OSS, & SQL.*

[9] Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. *ACM Computing Surveys, 41*(3), 1–58. https://doi.org/10.1145/1541880.1541882

[10] Kumar, S., Nandeshwar, S., & Kumar, N. (2023). Mechanism, tools, and techniques to mitigate distributed denial of service attacks. *International Journal for Research in Applied Science and Engineering Technology, 11*, 855–861. https://doi.org/10.22214/ijraset.2023.48675

[11] Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., & Yang, B. (2016). Predicting network attack patterns in SDN using a machine learning approach. In *Proceedings of the 2016 IEEE Conference on Network Function Virtualization and*

Software Defined Networks (NFV-SDN) (7–10). IEEE. https://doi.org/10.1109/NFV-SDN.2016.7919495

[12] Kodati, S., Vaishali, K., Gachikanti, S., Dastagiraiah, C., & Sreekanth, N. (2022). Design of an ensemble learning method for detection of distributed denial of service attacks in SDN using machine learning techniques. *Optics Communication and Networking*, *14*, 248–259. https://doi.org/10.1364/josac.589634

[13] Tzagkarakis, G., Ntouskos, A., & Ioannidis, S. (2019). A machine learning framework for early detection of DoS attacks in IoT-edge networks. In *2019 IEEE Global Communications Conference (GLOBECOM)* (1–6). IEEE. https://doi.org/10.1109/GLOBECOM38437.2019.9013945

[14] Liu, Y., Tang, J., He, X., & Zhong, Y. (2022). A two-tier DDoS attack detection strategy for SDN environments. *IEEE Transactions on Network and Service Management*, *19*(1), 101–113. https://doi.org/10.1109/TNSM.2021.3123504

[15] Salim, F. B., Shafiullah, G., Khan, M. K., & Islam, N. (2021). Deep learning-based framework for vulnerability detection and mitigation of DDoS attacks on IoT devices in smart grids. *Sensors*, *21*(12), 4140. https://doi.org/10.3390/s21124140\

[16] Hussain, F., Islam, N., Khan, M. K., & Al-Saggaf, A. M. (2020). A convolutional neural network-based approach for DDoS attack detection in 4G-LTE networks. *IEEE Access*, *8*, 151244–151255. https://doi.org/10.1109/ACCESS.2020.3017242

[17] Sahi, N., Chen, Z., & Tianfield, H. (2017). A deep learning approach for classification and mitigation of DDoS attacks in cloud computing environments. In *2017 International Conference on Cloud Computing and Big Data (ICCCBD)* (202–209). IEEE. https://doi.org/10.1109/ICCCBD.2017.8386642

[18] Ozer, E., Purohit, S., & Agrawal, A. (2017). An efficient deep learning-based approach for fast and accurate DDoS attack detection. In *2017 IEEE International Conference on Big Data (Big Data)* (4497–4504). IEEE. https://doi.org/10.1109/BigData.2017.8258504

[19] Osanaiye, O., Cai, H., Choo, K. K. R., Dehghantanha, A., Xu, Z., & Dlodlo, M. (2016). Ensemble-based multi-filter feature selection method for DDoS detection in cloud computing. *EURASIP Journal on Wireless Communications and Networking*, 130, https://doi.org/10.1186/s13638-016-0623-3

[20] Almadhor, A., Altalbe, A., Bouazzi, I., Al Hejaili, A., & Kryvinska, N. (2024). Strengthening network DDoS attack detection in heterogeneous IoT environment with federated XAI learning approach. *Scientific Reports*, *14*(1), 24322. https://doi.org/10.1038/s41598-024-76016-6

[21] Alahmadi, A. A., Aljabri, M., Alhaidari, F., Alharthi, D. J., Rayani, G. E., Marghalani, L. A., Alotaibi, O. B., & Bajandouh, S. A. (2023). DDoS attack detection in IoT-based networks using machine learning models: A survey and research directions. *Electronics*, *12*(14), 3103. https://doi.org/10.3390/electronics12143103

[22] Saravanan, R., Sangeetha, M., & Kavitha, B. (2023). Enhancing DDoS detection in SDIoT through effective feature selection and machine learning. *PLOS ONE*, *18*(11), e0309682. https://doi.org/10.1371/journal.pone.0309682