INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact Factor: 6.078
(Volume 11, Issue 5 - V11I5-1146)
Available online at: https://www.ijariit.com

A Comparative Study of Machine Learning Algorithms for Real-Time DDoS Detection in Cloud Environments

Oduwunmi Odukoya
odukoyao@etsu.edu
East Tennessee State University, Tennessee

Mariam Adetoun Sanusi
Sanusiadetoun@gmail.com
University of Texas, Dallas

Samuel Adenekan
adenekanp@etsu.edu
East Tennessee State University, Tennessee

ABSTRACT

Cloud environments are scalable and cost-effective, but they are also highly susceptible to cyberattacks, such as Distributed Denial of Service (DDoS) attacks, which can exhaust resources and impact availability. To counter these threats, this research investigates supervised machine learning techniques for identifying such attacks in real-time based on the BCCC cPacket Cloud DDoS 2024 dataset. Following deep preprocessing and exploratory data analysis, a multi-class classifier was employed to differentiate between benign, suspicious, and attack traffic. Of the models to be compared, the Decision Tree Classifier achieved the highest mark with an accuracy rate of 96.8 percent, which indicates its ability to categorize the majority of cases of traffic accordingly. The other models, including K Nearest Neighbors, Ridge Regression, Logistic Regression, and Linear Support Vector Classifier, had lower levels of accuracy, with Decision Tree always delivering the best. The results confirm that the Decision Tree is the best and most efficient model for precise real-time identification of DDoS attacks in cloud systems.

Keywords: *DDOS, Cloud Environment, Machine Learning, Cyber-attack.*

1. INTRODUCTION

Cloud environments have transformed contemporary computing through their enhancement of scalability, elasticity, and cost efficiency, making possible the foundation of digital services in industries (Sarker et al., 2022). Unfortunately, their distributed nature exposes them to serious security attacks, where the most common one is Distributed Denial of Service attacks, which overwhelm network and computational resources, causing service unavailability and financial loss (Islam et al., 2022).

These attacks are especially difficult in cloud environments, as resources are distributed dynamically and high traffic can hide malicious activity (Alashhab et al., 2022). Traditional detection methods, such as signature-based and rule-based detection, are usually unable to match the changing tactics of attacks and continuously improving adversaries (Mihoub et al., 2022). These traditional approaches are static and cannot offer accurate, real-time detection in real-time cloud environments, necessitating the adoption of more intelligent, dynamic solutions (Almaraz-Rivera et al., 2022). Machine learning has emerged as a viable solution to this problem with its capacity to learn from vast traffic data, recognize sophisticated attack patterns, and evolve to counter nascent threat forms (Alduailij et al., 2022).

Unlike static detection systems, machine learning models are capable of generalizing from past experiences to identify and anticipate Distributed Denial of Service attacks more accurately and effectively (Liu et al., 2022). Traditional supervised models and deep learning strategies have been proven in recent studies to be feasible for detecting network traffic abnormalities across various environments (Ramapatruni et al., 2019). Specifically, real-time detection with the application of machine learning allows for benign, suspect, and malicious traffic to be categorized with almost zero latency, which is important in the prevention of attacks before impacting system performance (Islam et al., 2022).

This paper is a comparative study of the five supervised machine learning models, i.e., Decision Tree, K Nearest Neighbors, Logistic Regression, Ridge Regression, and Linear Support Vector Machine, for real-time identification of Distributed Denial of Service attacks against cloud infrastructure.

The models are contrasted utilizing the BCCC cPacket Cloud DDoS 2024 dataset, which offers end-to-end traffic traces needed for strong training and validation (Lee et al., 2023).

An accurate preprocessing procedure guarantees the stability and accuracy of the data set, and exploratory data analysis discovers patterns and relationships between features that are essential for successful classification (Matsuzaka & Uesawa, 2023). Scikit learn is used for model deployment and testing using metrics such as accuracy, precision, recall, F1 score, inference time, and confusion matrix analysis to offer a comprehensive performance metric (Jamieson et al., 2023b).

The importance of this study is that it is a significant contribution to enhancing cloud security by systematic comparison of machine learning methods that achieve prediction accuracy and retain computational efficiency. Through the determination of the best model, especially in real-time scenarios, this study enables more intelligent detection systems with a capability to protect cloud systems against constantly changing Distributed Denial of Service attacks (Hendren et al., 2022). Along the journey, it not only emphasizes the role of machine learning in contemporary cybersecurity but also provides hands-on tips about how to choose algorithms that can provide consistent, timely, and scalable security in real-world cloud environments.

2. LITERATURE REVIEW

This literature review examines the application of machine learning to enhance the identification of Distributed Denial of Service (DDoS) attacks in cloud computing, whose scalability and shared environment make the infrastructure extremely susceptible to these attacks. The essay goes ahead to address traditional methods that include rule-based and signature-based solutions as no longer adequate in countering changing and advanced pattern attacks due to their inflexibility and failure to yield accuracy when applied in dynamic environments. Therefore, instead of depending on conventional methods that are not timely and exact, scientists have been seeking machine learning methods capable of handling vast traffic data, revealing intrinsic patterns, and presenting exact predictions in real-time. The following-cited papers are valuable contributions in such research, such as novel methods, achievements in performance, and loopholes calling for future innovation in cloud services' real-time security systems.

According to AlSaleh et al. (2024), *A DDoS Cloud Attacks Detection and Classification Framework Using Bayesian Convolutional Neural Network* came
up with a novel method that combines Bayesian inference and convolutional neural networks to identify Distributed Denial of Service attacks in cloud-based systems.

In addition to classifying benign and attack traffic, uncertainty estimates are also produced to enable improved decision-making in real-world applications. Experimental performance showed that Bayesian CNN posted a remarkable accuracy of 99.66 percent, far superior to existing detection models. The strongest aspect of the work is that it excels in multi-class classification, but there is a vast disparity in the evaluation for different cloud providers, restricting its generalizability in real-world multi-tenant deployments.

According to Songa et al. (2024), an *Integrated SDN DDoS Attack Detection/Defense and Anomaly Primary Cause Localization Framework for Cloud Computing* proposed a sophisticated Software Defined Networking-based detection and defense technique. The framework unites anomaly detection and anomaly root-cause localization, facilitating prevention and rapid response to Distributed Denial of Service attacks. Comparative comparison of machine learning classifiers showed that the highest performing configuration reached a high accuracy of 99.92 percent. This proves the effectiveness of SDN-based defenses in cloud environments. The research also identified a potential limitation in the use of SDN controllers, which would be a potential point of failure, and experiments were limited to certain cloud environments; hence, external validity was limited.

According to Bamasag et al. (2022), *RT-EDyNet: A Multi-Agent Real-Time DDoS Attack Monitoring and Detection System for Cloud Computing Environments* proposed a cooperative and distributed detection based on various monitoring agents in the cloud. Various machine learning classifiers have been applied to cloud traffic, and Random Forest was the best performing with an accuracy rate of 99.38 percent. The system proved to be functional in real-time attack monitoring, proving that a distributed agent-based architecture is scalable with cloud systems. Nevertheless, studies found there was no benchmarking between adversarial attack testing and cloud service providers, something that would possibly make it difficult for the system to adjust.

According to Awan et al., 2021, *Real-Time DDoS Attack Detection System Using Big Data Approach* designed a scalable detection pipeline based on Apache Spark to solve high-speed cloud traffic streams. Several classifiers were tried under the big data system, and the optimal model was created to obtain 99.5 percent accuracy while being low in latency. This outcome highlights the strength of big data systems in identifying large-scale detection. Despite showing remarkable results, the study was tainted by a lack of cross-dataset validation and an inability to focus enough on low-rate stealth attacks that continue to pose a daunting challenge in contemporary Distributed Denial of Service defence. According to Alduailij et al. (2022), *Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method* emphasized the essence of feature extraction in ensuring model performance for cloud traffic classification. Through the use of mutual information and Random Forest feature importance, authors reduced dimensions and increased detection efficiency. Random Forest and ensemble learning algorithms obtained about 99 percent accuracy, depicting the advantage of feature selection targeted towards particular characteristics. Nonetheless, the limitation was also the use of binary classification, limiting the model from handling nuanced traffic categories, as well as testing insufficiency when there are changes in cloud traffic patterns due to concept drift.

According to Bhardwaj et al. (2020), a *Hyperband Tuned Deep Neural Network With Well-Posed Stacked Sparse Autoencoder for Detection of DDoS Attacks in Cloud* applied deep learning methods using stacked sparse autoencoders and Hyperband hyperparameter search for optimizing detection performance.

The deep neural network produced over 99 percent accuracy on cloud DDoS datasets and was among the top-performing methods. Although successful, the model's lack of interpretability and complexity are concerns for real-time deployment within the cloud, and its resilience to noisy multi-tenant environments was not exhaustively considered.

This literature review highlights that machine learning algorithms always attain high accuracies in identifying Distributed Denial of Service attacks in cloud systems, usually more than 99 percent performance on a variety of datasets and experimental configurations.

Although the best high-performance models include Random Forest, Bayesian CNN, and deep neural networks, all the surveyed papers document shortcomings in cross-cloud validation, dynamic low-rate attack pattern adaptability, and interpretability towards practical implementation. Such shortcomings remind us to pursue further research on large-scale, interpretable, and robust detection mechanisms that can defend existing cloud infrastructures from ever-changing attack modes.

3. METHODOLOGY

This methodology describes the systematic steps taken to conduct a comparative study of machine learning models for real-time Distributed Denial of Service detection in cloud computing networks. The major aim was to critically analyze the performance of five supervised learning models, i.e., Decision Tree, K Nearest Neighbors, Logistic Regression,
Ridge Regression, and Linear Support Vector Machine, on the BCCC cPacket Cloud DDoS 2024 data. The methodology was conscientiously designed to be accurate, consistent, and equitable at all steps of data preparation, analysis, model deployment, and appraisal.

3.1 About Dataset

The dataset used for this study is the BCCC cPacket Cloud DDoS 2024 dataset, a large collection of cloud traffic logs specifically tailored for Distributed Denial of Service detection research. The original data contained 540,494 rows and 319 columns that encode a broad array of numerical features defining numerous properties of network traffic behavior, with a single categorical label column to determine classes of benign, suspicious, and attack traffic. For enhancing quality and pertinence, extensive cleaning of data was done, which included removing duplicate attributes, fixing invalid values, verifying data types, and removing duplicate records. This processing reduced the dataset to 518,965 rows and 46 columns to only keep up with high-quality and relevant features for analysis. The distribution of discrete and continuous variables in the data and the multi-class labeling is best suited for the data to be utilized for training and testing machine learning algorithms for real-time DDoS detection in the cloud.

3.2 Data Preprocessing

Data cleaning was performed to ensure that the data is accurate, reliable, and ready for the subsequent data analysis and model training steps.

First, the dataset was streamlined to focus only on the most relevant attributes by selecting a subset of the original columns of the dataset. Columns considered redundant, irrelevant, or unlikely to contribute meaningfully to the classification task were excluded. This targeted feature selection was guided by domain knowledge of network traffic analysis and DDoS detection, and complemented by exploratory data analysis, ensuring that only variables with significant relevance and predictive value for distinguishing between attack, benign, and suspicious traffic were retained. This step not only reduced dataset dimensionality and computational complexity but also helped minimize the risk of overfitting, thereby improving the overall efficiency and effectiveness of the machine learning models.

Second, the dataset was examined for the presence of missing and infinite values, as these anomalies can distort statistical analysis and negatively impact model performance. However, no such values were detected, confirming that the dataset was complete and ready for subsequent preprocessing and modeling steps, without the need for additional imputation procedures.

In addition, the dataset was checked to identify columns containing negative values. Several features were found to contain negative entries. For time-based attributes (durations), negative values are not logically valid sincetime cannot be less than zero. Consequently, these values were converted to their absolute equivalents, ensuring that all time-based features represented realistic, non-negative durations. In contrast, delta-related features, which represent the difference between two measurements, can legitimately assume negative values, as the magnitude and direction of change may vary. Thus, these values were retained without modification to avoid distorting potentially meaningful variations in the data. This correction preserved the integrity of the dataset and maintained consistency for subsequent analysis and model training.

Moreover, the data types of all columns were carefully examined to ensure they were correctly assigned. This verification step is essential to prevent type-related inconsistencies that could affect calculations or model training. The inspection confirmed that each feature had the appropriate data type, thereby enabling accurate processing, analysis and interpretation.

Additionally, the dataset was examined for single-valued columns, i.e., features containing the same value for all entries. Such features provide no variability and thus contribute no discriminative power to the model. No single-valued columns were found in the dataset, ensuring that all retained features offered potential informational value for pattern detection and classification tasks.

Finally, several duplicate rows were found in the dataset. These are records containing identical values across all features. If left unaddressed, these entries could over-represent certain patterns and introduce bias. Thus, the duplicate entries were removed to ensure a cleaner, more balanced dataset, ensuring that model training was based on accurate and unbiased information.

3.3 Exploratory Data Analysis

The dataset used in this study is the BCCC-cPacket-Cloud-DDoS-2024 dataset (available on Kaggle).

Initially, the dataset contained 540,494 rows and 319 columns. Following a rigorous data cleaning process, the dataset was reduced to 518,965 rows and 46 columns, ensuring that only relevant and high-quality attributes were retained for analysis.

Only the label column is categorical; it contains distinct class values that represent different types of cloud traffic. All other features are numerical, comprising both continuous and discrete measurements extracted from network traffic statistics.

Table 1 presents a summary of each column, including its data type, number of unique values, and range of observed values.

S/N	Dataset Column	Data type	Unique	Range
1.	src_port	Numeric	45330	2...65534
2.	dst_port	Numeric	65535	1...65535
3.	Duration	Numeric	23969	0...3570.693
4.	fwd_packets_count	Numeric	109	0...1097428
5.	bwd_packets_count	Numeric	110	0...2171786
6.	total_payload_bytes	Numeric	685	0...3166384000
7.	fwd_total_header_bytes	Numeric	207	0...35117700
8.	bwd_total_header_bytes	Numeric	203	0...53570600
9.	avg_segment_size	Numeric	840	0...1404.121
10.	fwd_init_win_bytes	Numeric	1822	0...65535
11.	bwd_init_win_bytes	Numeric	956	0...65535
12.	active_mean	Numeric	71	0...454.6559
13.	idle_mean	Numeric	108	0...299.9911
14.	bytes_rate	Numeric	2296	0...4555014000
15.	packets_rate	Numeric	25661	0...4194304
16.	down_up_rate	Numeric	274	0...38
17.	avg_fwd_bulk_rate	Numeric	208	0...9110028000
18.	avg_bwd_bulk_rate	Numeric	136	0...417824100
19.	fwd_bulk_state_count	Numeric	26	0...162
20.	bwd_bulk_state_count	Numeric	28	0...128
21.	fwd_bulk_total_size	Numeric	155	0...1569076000
22.	bwd_bulk_total_size	Numeric	119	0...7076706
23.	fwd_bulk_duration	Numeric	196	0...81.88203
24.	bwd_bulk_duration	Numeric	134	0...21.33963
25.	fin_flag_counts	Numeric	8	0...25
26.	psh_flag_counts	Numeric	109	0...262189
27.	urg_flag_counts	Numeric	2	0, 1
28.	ece_flag_counts	Numeric	14	0...20
29.	syn_flag_counts	Numeric	47	0...57
30.	ack_flag_counts	Numeric	142	0...2524043
31.	cwr_flag_counts	Numeric	12	0...18
32.	rst_flag_counts	Numeric	2	0, 1
33.	subflow_fwd_packets	Numeric	185	0...182904.7
34.	subflow_bwd_packets	Numeric	173	0...434357.2
35.	subflow_fwd_bytes	Numeric	317	0...261512700
36.	subflow_bwd_bytes	Numeric	317	0...261512700
37.	fwd_packets_IAT_mean	Numeric	4920	0...1703023000
38.	bwd_packets_IAT_mean	Numeric	3586	0...1703023000
39.	handshake_duration	Numeric	625	0.0002...8.2212
40.	handshake_state	Numeric	4	0...3
41.	delta_start	Numeric	395	0...150
42.	mean_packets_delta_time	Numeric	20944	0...999.999
43.	mean_packets_delta_len	Numeric	641	-1460...1460
44.	mean_header_bytes_delta_len	Numeric	158	-32...20
45.	mean_payload_bytes_delta_len	Numeric	568	-1448...1448
46.	label	Categorical	3	'Benign', 'Suspicious', 'Attack'

The categorical field 'label' serves as the target variable in this study. It categorizes cloud traffic instances into three distinct classes: Benign, Suspicious and Attack. Benign represents legitimate, non-malicious cloud traffic. Suspicious denotes anomalous traffic patterns that do not conform to normal behaviour and may indicate potential security threats. An attack refers to confirmed malicious traffic associated with DDoS activity.

This target variable is central to this classification task, where the objective is to accurately predict the class of a given network traffic instance based on patterns and characteristics derived from the other features in the dataset. The ability to correctly identify each class is critical for enabling real-time detection and mitigation of DDoS attacks in cloud environments.

The figure below illustrates the variation in the average number of backwards packets across the three traffic categories. Benign traffic demonstrates

the highest volume of backwards communication, consistent with the bidirectional data exchange expected in normal cloud operations. Suspicious traffic also shows a moderately high level of backwards packets, suggesting irregular but not entirely disrupted communication patterns.

In contrast, Attack traffic exhibits a substantially lower average, reflecting the minimal backwards communication typically associated with DDoS incidents, where the primary objective is to overwhelm the target rather than facilitate normal two-way communication.

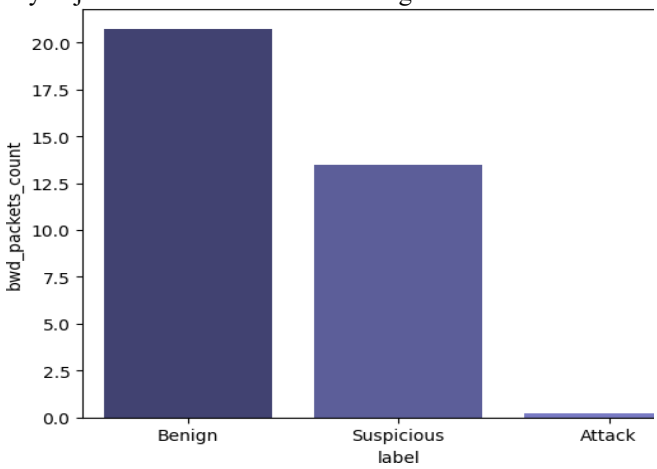


Figure 1: Average Backwards Packet Count by Traffic Category

Figure 2 presents the average size of the initial TCP window bytes for each traffic category. The initial TCP window bytes refers to the amount of data (in bytes) that a sender is permitted to transmit before receiving an acknowledgment from the receiver during the initial phase of a TCP connection.

Among the three classes, Attack traffic exhibits the highest average, indicating a larger initial TCP window allocation compared to benign and suspicious traffic. This may be indicative of attempts to maximize data transmission early in the connection

to rapidly overwhelm the target system.

Suspicious and Benign traffic display comparatively lower averages, reflecting more typical or restrained connection initiation patterns.

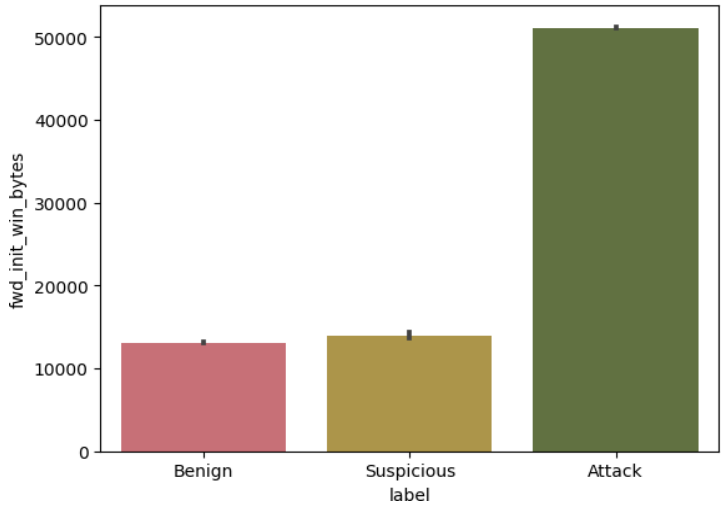


Figure 2: Average Initial TCP Window Size by Traffic Category

Figure 3 displays the distribution of forward header bytes for each traffic category. Forward header bytes represent the total size of header information, in bytes, sent in the forward direction of network communication.

Benign traffic exhibits the widest range, spanning roughly from 20 to around 50 bytes, which reflects the variability expected in normal, diverse cloud interactions. Suspicious traffic follows with a noticeably narrower range, generally between the 20 and about 30 bytes, suggesting less variation in header sizes for potentially anomalous connections.

Conversely, Attack traffic displays the smallest range overall, with most values clustered closely around its median, approximately 40 bytes, indicating more uniform header structures often characteristic of automated or scripted DDoS traffic.

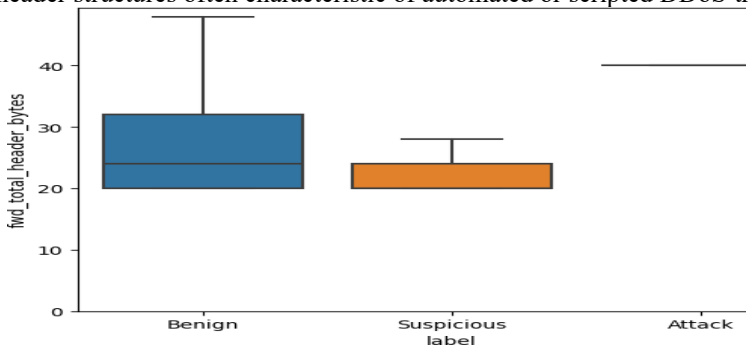


Figure 3: Distribution of Forward Header Bytes by Traffic Category.

Figure 4 shows the average Down/Up rate for each traffic category. The Down/Up rate measures the ratio between the amount of data received (downloaded) and the amount of data sent (uploaded) during a network connection.

Benign traffic has the highest average Down/Up rate; however, the value remains moderate, reflecting the responsive data exchanges typical of normal cloud operations.

Conversely, Attack traffic exhibits the lowest average Down/Up rate, consistent with DDoS activity, where incoming requests far exceed outgoing responses due to the target being overwhelmed or unresponsive.

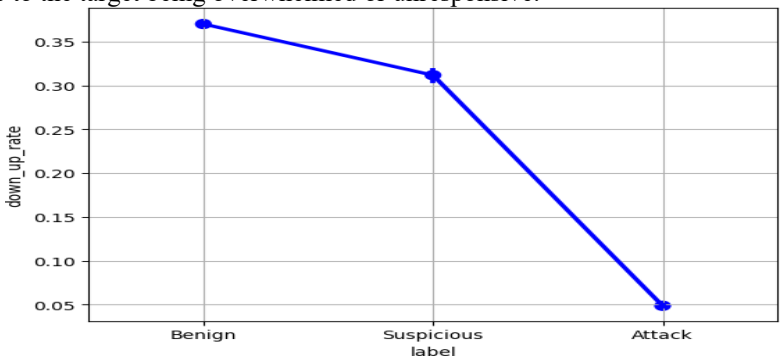


Figure 4: Average Down/Up Rate by Traffic Category.

Figure 5 depicts the maximum handshake duration observed for each traffic category. The handshake duration refers to the time required to complete the TCP three-way handshake process, which establishes a reliable connection between a sender and a receiver before data transfer begins. Benign traffic exhibits the longest maximum handshake duration, reflecting normal variations in network latency and connection setup times.On the other hand, Attack and Suspicious traffic have much shorter maximum handshake durations, likely because many connection attempts are incomplete or aborted, a common characteristic of DDoS attacks where rapid requests are made without fully establishing reliable connections.

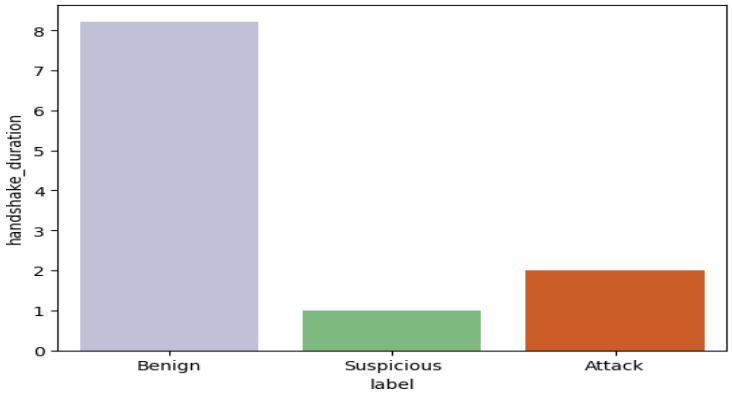


Figure 5: Maximum Handshake Duration by Traffic Category.

Figure 6 presents the average change in packet length between consecutive packets for each traffic category. This metric captures how the size of packets varies from one transmission to the next. Notably, Attack traffic is the only category exhibiting a positive change in packet length, indicating that packet sizes tend to increase slightly from one packet to the next. This pattern reflects the automated nature of DDoS attacks, where packets are systematically generated by scripts or bots to maximize disruption.

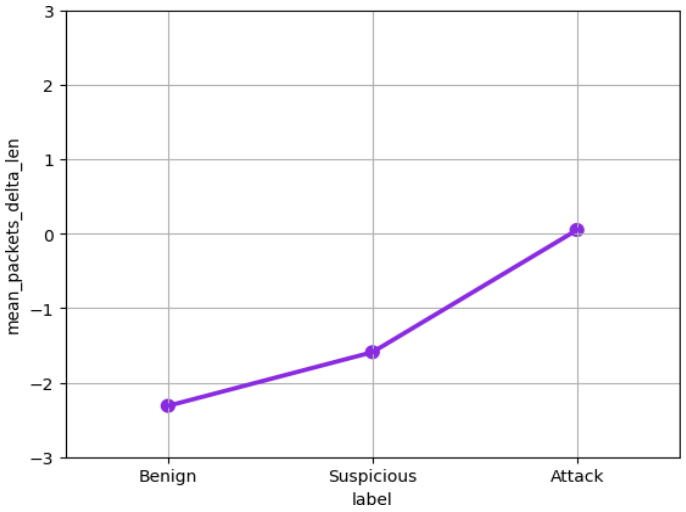


Figure 6: Average Change in Packet Length by Traffic Category.

Figure 7 illustrates the average count of ACK flags for each traffic category. The ACK (acknowledgment) flag in TCP packets indicates successful receipt of data and is essential for reliable communication.

Attack traffic shows a markedly low average count of ACK flags compared to Benign and Suspicious traffic. This reflects the incomplete connections typical of DDoS attacks, where responses from the target are minimal or absent, resulting in minimal acknowledgment of received packets. On the other hand, Benign traffic shows a substantially higher average, indicating normal, bidirectional communication and proper acknowledgment of received packets. This stark contrast highlights the potential of ACK flag patterns as a discriminative feature for distinguishing attack traffic from legitimate cloud traffic.

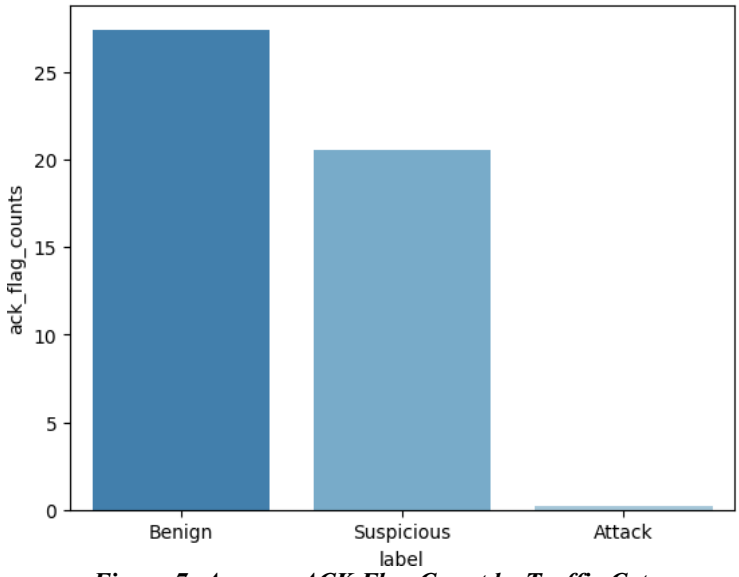


Figure 7: Average ACK Flag Count by Traffic Category

3.4 Label Transformation

Feature selection was conducted as part of the data cleaning phase. Since all features in the dataset are numeric, no additional encoding was required for the input variables. However, the target variable, label, is categorical and contains non-numeric values, which cannot be directly used in most machine learning algorithms.

To address this, a new column, label_num, was created. In this column, numerical codes were assigned to represent each class: 0 for Benign, 1 for Suspicious, and 2 for Attack. This transformation enabled the target variable to be directly incorporated into model training while preserving the categorical distinctions between traffic types.

See Figure 8 for the distribution of cloud traffic instances across these three categories.

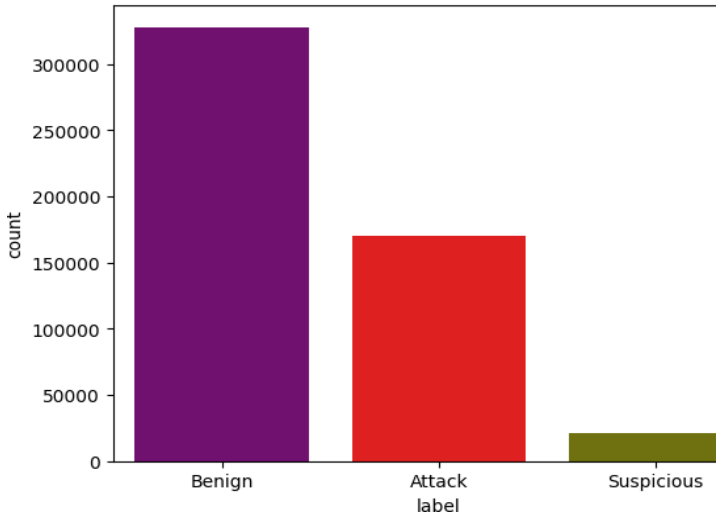


Figure 8: Class Distribution of the Target Variable

4. MODEL DEVELOPMENT AND EVALUATION

Prior to running machine learning algorithms or other stochastic processes, a fixed random seed value of 21 was set to ensure reproducibility and consistent results across multiple runs of the code.

The dataset was then split into two subsets: a training set comprising 70% of the data and a testing set comprising the remaining 30%, with stratification based on the target variable to preserve class proportions. The training set, being the larger portion of the dataset, was used to train the machine learning models. During this process, the models learn patterns and relationships from the training data in order to make accurate classifications.

The machine learning algorithms employed in this study include Decision Tree, K-Nearest Neighbors, Logistic Regression, Ridge Regression, and Linear Support Vector Machine. After training, the models were evaluated on the test set using relevant evaluation metrics, allowing for an assessment of
their generalization ability on unseen data.

The results of this evaluation are presented in the table below, allowing for a comparison of how well each classifier performed on this task.

Note: The precision, recall, and F1-score in the table below are computed using the macro-averaging approach. This approach calculates the metric for each class individually and then takes the unweighted average across all classes, treating each class equally regardless of its size. By giving equal weight to each class, macro averaging ensures that the evaluation metrics reflect the model's performance across all classes, providing a more balanced and informative assessment of the models' effectiveness in detecting Benign, Suspicious, and Attack traffic.

Table 2: Evaluation Results

Model	Accuracy	Precision	Rec
Decision Tree Classifier	0.968	0.888	0.88
K-Neighbors Classifier	0.954	0.875	0.8
Logistic Regression	0.867	0.570	0.5
Ridge Classifier	0.877	0.661	0.6
Linear SVC	0.662	0.546	0.5

Figure 9 presents the accuracy scores of the different machine learning models evaluated on the test set. Among the classifiers, the Decision Tree Classifier achieved the highest accuracy, demonstrating strong predictive performance and its ability to capture the underlying patterns in the dataset. K-Neighbors Classifier also performed well, reflecting its effectiveness in classifying instances based on similarity in feature space.

Ridge Classifier and Logistic Regression showed moderate accuracy, indicating that while they can capture some of the relationships in the data, their linear assumptions may limit performance in this multi-class DDoS detection task. In contrast, the Linear Support Vector Classifier (Linear SVC) recorded the lowest accuracy, suggesting that it is less suitable for this dataset.

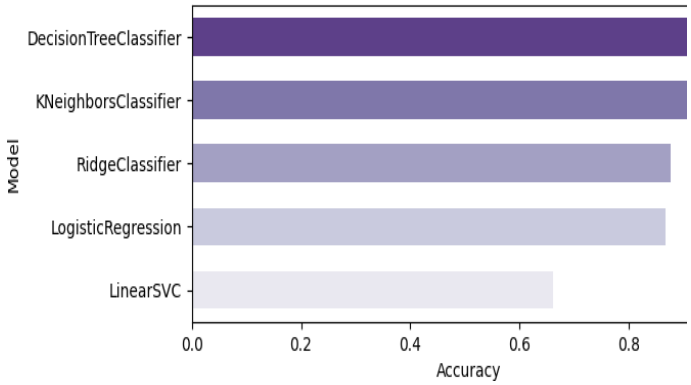


Figure 9: Accuracy Scores of Machine Learning Models.

Figure 10 shows the macro precision scores of the evaluated models. The Decision Tree Classifier achieved the highest macro precision, demonstrating strong capability in making correct predictions across all traffic categories. On the other hand, the Linear Support Vector Classifier recorded the lowest macro precision, suggesting that it frequently misclassifies instances, and is therefore less effective for distinguishing between the classes in this DDoS detection task.

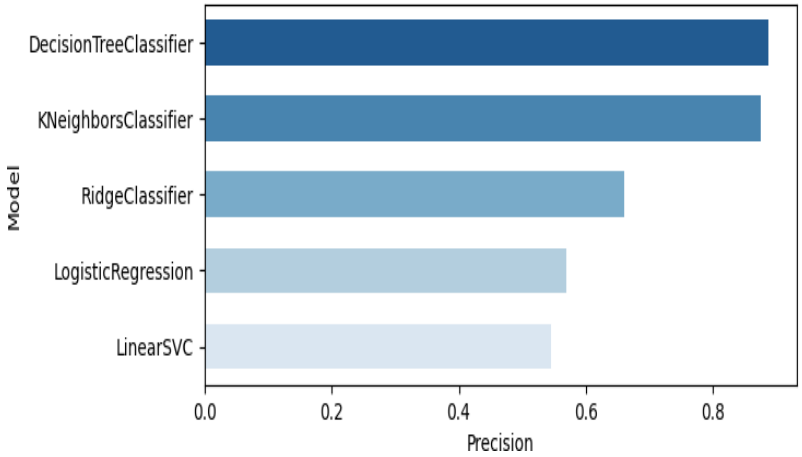


Figure10: Macro Precision Scoresof Machine Learning Models

Figure 11 presents the macro recall scores of the evaluated machine learning models. The Decision Tree Classifier achieved the highest macro recall, indicating its strong ability to correctly identify instances from all traffic categories. The Linear Support Vector Classifier recorded the lowest macro recall, suggesting its limited effectiveness in detecting all relevant instances in this multi-class DDoS detection task.

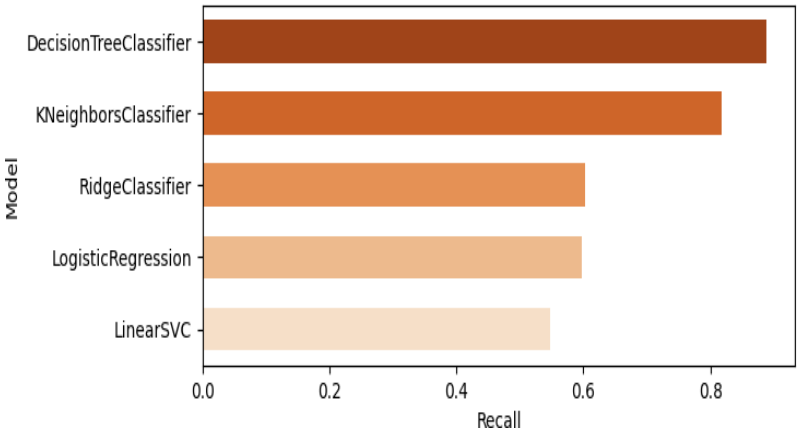


Figure11: Macro Recall Scores of Machine Learning Models.

Figure 12 presents the macro F1-scores of the evaluated machine learning models. The Decision Tree Classifier achieved the highest macro F1-score, demonstrating a strong balance between precision and recall across all traffic categories. The Linear Support Vector Classifier recorded the lowest macro F1-score, reflecting its limitations in capturing the full complexity of the multi-class DDoS detection task.

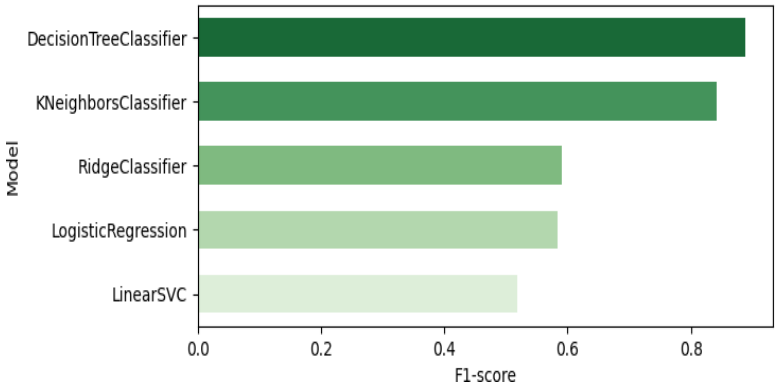


Figure 12: Macro F1-Scores of Machine Learning Models.

Figure 13 shows the inference times of the evaluated machine learning models, measured as the time taken to make predictions on the test set. Logistic Regression exhibited the fastest inference time, followed closely by Ridge Classifier and Linear SVC, indicating their suitability for real-time deployment. The Decision Tree Classifier also performed efficiently, with slightly longer inference time. In contrast, K-Nearest Neighbors showed a significantly higher inference time, reflecting the computational cost of comparing new instances to all training samples during prediction, which
may limit its practicality in real-time DDoS detection scenarios.

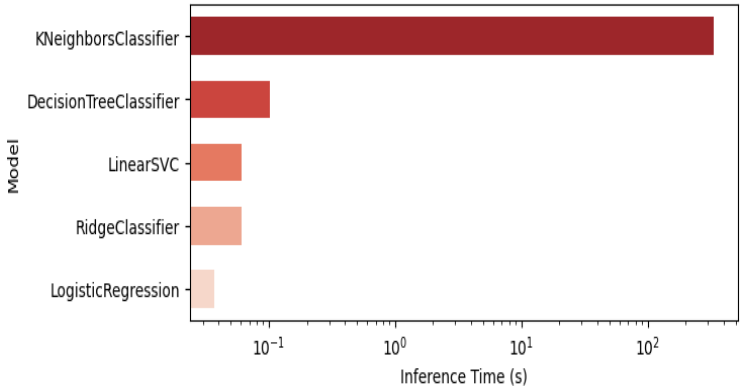


Figure 13: Inference Times of Machine Learning Models

Tables 3 to 7 present the classification reports for each machine learning model. Each report provides a detailed summary of the model's performance across all traffic categories, including key metrics such as precision, recall, and F1-score. These reports show the strengths and limitations of each model in detecting Benign, Suspicious, and Attack traffic.

Table 3: Classification Report of Decision Tree Classifier

Label	Precision	Recall	F1-score
Benign	0.99	0.99	0.99
Suspicious	0.72	0.72	0.72
Attack	0.96	0.96	0.96

From the table above, the classification report for the Decision Tree Classifier shows that the model achieved very high performance on the Benign and Attack classes, with precision, recall, and F1-score all close to 0.99 and 0.96, respectively, indicating that it correctly identifies most instances in these categories.
The Suspicious class shows lower performance, with precision, recall, and F1-score around 0.72, suggesting that the model finds it more challenging to distinguish anomalous traffic from other classes.

Table 4: Classification Report of K-Neighbors Classifier

Label	Precision	Recall	F1-score
Benign	0.97	0.98	0.97
Suspicious	0.71	0.52	0.60
Attack	0.95	0.95	0.95

From the table above, the classification report for the K-Neighbors Classifier shows that the model performs well on the Benign and Attack classes, achieving high recall of 0.98 and 0.95, which indicates reliable classification of these categories. However, its performance on the Suspicious class is noticeably lower, particularly in recall (0.52), suggesting that the model struggles to correctly identify all instances of anomalous traffic.

Table 5: Classification Report of Logistic Regression

Label	Precision	Recall	F1-score
Benign	0.90	0.92	0.91
Suspicious	0.00	0.00	0.00
Attack	0.81	0.87	0.84

From the table above, the classification report for the Logistic Regression model shows that the model performs reasonably well on the Benign class, achieving precision and recall of 0.90 and 0.92 respectively, indicating reliable identification of normal traffic instances. Its performance on the Attack class is moderate, with precision, recall, and F1-score between 0.81 and 0.87, showing that the model can detect most DDoS traffic. In contrast, the model completely fails to classify the Suspicious class, with precision, recall, and F1-score all at 0.00. This suggests that Logistic Regression model is unable to detect anomalous traffic in this dataset, likely due to the complexity of the class boundaries or class imbalance.

Table 6: Classification Report of Ridge Classifier

Label	Precision	Recall	F1-score
Benign	0.90	0.93	0.92
Suspicious	0.25	0.00	0.00
Attack	0.84	0.87	0.85

The classification report for the Ridge Classifier shows that the model performs well on the Benign class, achieving precision, recall, and F1-score around 0.90–0.93, indicating strong identification of normal traffic instances. For the Attack class, performance is moderate, with precision and recall scores of 0.84 and 0.87, showing that the model identifies most DDoS traffic.

However, the model struggles with the Suspicious class, achieving very low precision and zero recall and F1-score, which highlights its inability to correctly identify anomalous traffic.

Table 7: Classification Report of Linear Support Vector Classifier

Label	Precision	Recall	F1-score
Benign	0.92	0.59	0.72
Suspicious	0.22	0.19	0.20
Attack	0.51	0.86	0.64

The classification report for the Linear Support Vector Classifier shows that the model achieves moderate performance on the Benign class, with a precision of 0.92 but a recall of only 0.59, indicating that while predictions are often correct, many normal traffic instances are missed. The Attack classifier has a higher recall (0.86) but lower precision (0.51), suggesting that the model identifies most attack instances but also produces several false positives. The Suspicious class shows the poorest performance, with low precision, recall, and F1-score around 0.20, highlighting the model's difficulty in detecting anomalous traffic.

Overall, Benign traffic is consistently the easiest class to identify across all models. Most classifiers achieve high precision and recall for Benign instances, indicating that normal cloud traffic exhibits stable and consistent patterns that the models can reliably detect. Suspicious traffic, however, poses the greatest challenge for all classifiers. Precision, recall, and F1-scores are consistently low for this class, reflecting the difficulty in correctly identifying anomalous traffic. This may be due to its intermediate characteristics, which overlap with both Benign and Attack patterns, as well as the smaller number of Suspicious instances in the dataset.

4.2 Confusion Matrices

Figure 14(a–e) presents the confusion matrices for the evaluated models, illustrating how each model predicted the different cloud traffic categories. These matrices provide a visual summary of the models' performance for each class, showing the number of correctly classified instances along the diagonal and misclassifications in the off-diagonal entries.

The Decision Tree Classifier shows strong overall performance but incorrectly classified 1,260 Benign instances, 1,746 Suspicious instances, and 2,007 Attack instances.

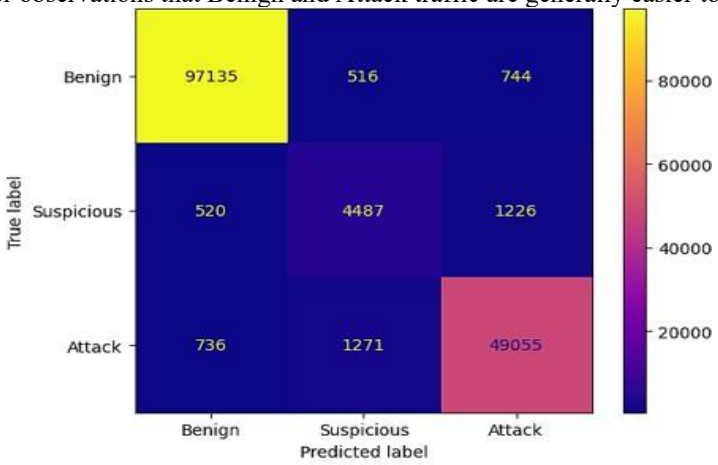
The K-Nearest Neighbors Classifier misclassified 1,665 Benign instances, 3,001 Suspicious instances, and 2,528 Attack instances.

The Logistic Regression classifier inaccurately classified 7,978 instances of Benign traffic and 6,505 instances of Attack traffic. It completely failed to identify any Suspicious instances, which were all classified as either Benign or Attack traffic.

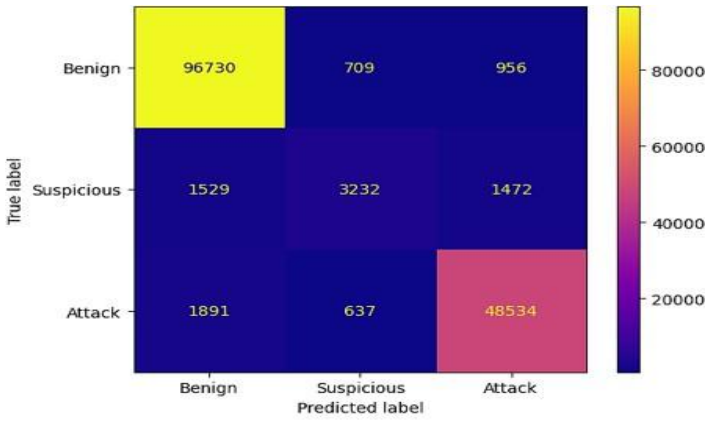
The Ridge Classifier mislabeled 6,497 Benign instances and 6,468 Attack instances while detecting only a single instance of Suspicious traffic.

The Linear Support Vector Classifier exhibited the poorest performance overall, misclassifying 40,593 Benign instances, 5,034 Suspicious instances, and 7,042 Attack instances.

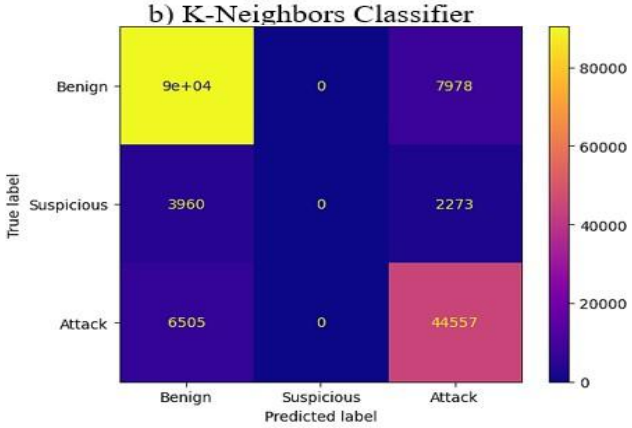
These results reinforce the earlier observations that Benign and Attack traffic are generally easier to classify than Suspicious traffic.



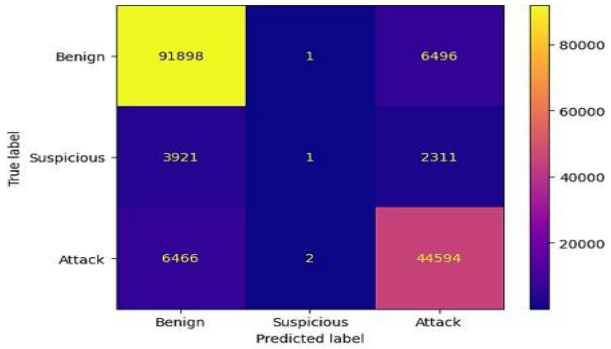
a) Decision Tree Classifier



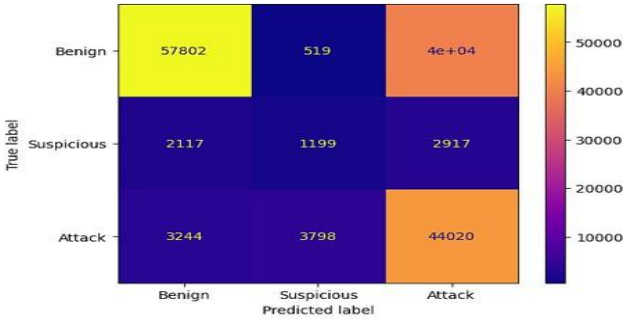
b) K-Neighbors Classifier



c) Logistic Regression



d) Ridge Classifier



e) Linear Support Vector Classifier

Figure 14 (a-e): Confusion Matrices of Machine Learning Models

5. DISCUSSION OF RESULTS

The Decision Tree Classifier demonstrated the highest overall performance among the evaluated machine learning models. It achieved an accuracy of 96.8%, indicating that it correctly classified the vast majority of instances in the test set. Its macro precision, recall, and F1-score were all approximately 88.9%, reflecting strong and balanced performance across all three traffic categories– Benign, Suspicious, and Attack. These results indicate that the model is effective at capturing the distinguishing patterns in the dataset while minimizing misclassifications. Moreover, the Decision Tree Classifier exhibited a fast inference time of 0.1029 seconds, highlighting its suitability for real-time DDoS detection where timely predictions are critical.

The K-Nearest Neighbors Classifier achieved a strong performance with an overall accuracy of 95.4%, correctly classifying most instances in the test set. Its macro precision, recall, and F1-score were approximately 87.5%, 81.7%, and 84.1%, respectively, reflecting good performance across all three traffic categories. One notable limitation of this model is its high inference time of 330.6965 seconds, which is significantly longer than that of the other models due to the computational overhead of comparing each test instance to all training samples. This implies that its practical deployment for real-time DDoS detection may be constrained by computational efficiency.

The Ridge Classifier achieved an overall accuracy of 87.7%, indicating moderate performance in correctly classifying the test set instances. Its macro precision, recall, and F1-score were 66.1%, 60.2%, and 59.0%, respectively, reflecting moderate effectiveness across all three traffic categories, particularly for the minority Suspicious class. Despite these limitations, the Ridge Classifier benefits from a relatively fast inference time of 0.0604 seconds, making it suitable for scenarios that require timely predictions.

The Logistic Regression model achieved an overall accuracy of 86.7%, indicating moderate performance in correctly classifying instances in the test set. Its macro precision, recall, and F1-score were 57.0%, 59.7%, and 58.3%, respectively, reflecting a significant drop in balanced performance across the three traffic categories, with zero precision and recall for the Suspicious traffic category. While the model performs reasonably well in identifying Benign and Attack traffic, it struggles to detect Suspicious instances, likely due to overlapping patterns and class imbalance. However, Logistic Regression exhibited the fastest inference time of 0.0373 seconds, demonstrating high computational efficiency and suitability for real-time applications, although its predictive accuracy for this DDoS detection task is limited.

The Linear Support Vector Classifier demonstrates limited utility for effective DDoS detection. It recorded an overall accuracy of 66.2%, indicating relatively low performance in correctly classifying instances in the test set. Its macro precision, recall, and F1-score were 54.6%, 54.7%, and 51.9%, respectively, reflecting limited effectiveness across the three traffic categories. While the Linear SVC maintains a fast inference time of 0.0606 seconds, which is suitable for real-time applications, its predictive accuracy is significantly lower compared to other evaluated models.

Considering the above results, the Decision Tree Classifier outperforms all other machine learning models used in this study, making it the most suitable choice for this DDoS detection task. It achieved the highest overall accuracy of 96.8%, indicating that it correctly classified the vast majority of instances in the test set. Its macro precision, recall, and F1-score are all approximately 88.9%, demonstrating consistent and reliable performance across all three traffic categories. It also had the lowest number of misclassifications. The Decision Tree Classifier also exhibits a fast inference time of 0.1029 seconds, making it highly suitable for real-time DDoS detection in a cloud environment. This combination of accuracy, balanced class performance, and computational efficiency justifies its selection as the most effective and efficient machine learning model for real-time detection of DDoS threats in cloud environments.

6. RESEARCH GAP

This research directly addresses the gaps identified in existing studies in the context of existing efforts across the literature review and augments the contribution of DDoS detection in cloud environments. AlSaleh et al. (2024) achieved incredibly high accuracy with a Bayesian CNN but conceded an evaluation gap across heterogeneous cloud vendors as escalating issues towards application practicability. Also, Songa et al. (2024) designed an SDN-based architecture with 99.92 percent accuracy but with the shortcoming of relying on SDN controllers that establish single points of failure. Bamasag et al. (2022) designed an agent-based architecture that was scalable but with no benchmarking against various providers and adversary techniques. In comparison to these studies, the current work offers a different solution in terms of computational efficiency and multi-class classification for a real dataset with a distinction between attack, suspicious, and benign traffic. The highest-rated algorithm, Decision Tree Classifier, achieved 96.8 percent accuracy with balanced precision, recall, and F1-measures of 88.9 percent and was also found to be strong for all three categories of traffic. Above all, its 0.1029-second high-speed inference affords real-world practicability for real-time detection, closing the most critical efficiency and flexibility chasms neglected or under-emphasized in existing work.

It also builds and extends Awan et al. (2021), Alduailij et al. (2022), and Bhardwaj et al. (2020) contributions in terms of methodology. Awan et al. detected scalability with Apache Spark but didn't properly describe stealthy low-rate attacks, while Alduailij et al. demonstrated the advantage of feature selection without using binary classification solely. Bhardwaj et al. advanced detection to over 99 percent accuracy using deep neural networks, but added higher complexity and reduced interpretability and thereby made real-time deployment in multi-tenant networks difficult.

This paper surpasses such constraints by employing a multi-class classification technique that allows it not only to identify attack traffic but also suspicious traffic, which is necessary for proactive defense against low-rate and adaptive attacks. Also, with the use of interpretable models such as Decision Trees accompanied by comparative analysis with more basic techniques such as Logistic Regression and Ridge Regression, this research strikes a balance between accuracy, scalability, and interpretability. In doing so, it bridges the existing gaps among the discussed research works by providing a detection mechanism that is accurate, efficient, interpretable, and tunable and thus better poised to be deployed within dynamic cloud environments in reality, whose attack methods often keep altering.

7. CLOUD CYBERATTACK DETECTION USING DECISION TREE

This research confirms that the Decision Tree Classifier is an extremely effective technique for identifying Distributed Denial of Service attacks on cloud networks. It recorded the highest accuracy of 96.8 percent compared to all the models that it was benchmarked against, with highly balanced macro precision, recall, and F1-scores for benign, suspicious, and attack traffic classes. The model's capacity to accurately identify non-linear traffic patterns without any compromise on inference time makes it suitable for real-time operation in cloud settings. Notably, the Decision Tree is also explainable, and security developers can understand how features lead towards classifications, importantly in cybersecurity spaces where interpretability increases trust and usability (Lee et al., 2023).

The effort in this research contributes further to cybersecurity developers and academic researchers. There has been high accuracy using previous research work with models like ensemble learning and deep neural networks, but most of the methods were criticized for excessive computational overhead and insensitivity in real-time settings (Matsuzaka and Uesawa, 2023). On the other hand, the Decision Tree model described here achieves high accuracy without this complexity and hence breaks the trade-off between accuracy and efficiency. Anomaly detection models for smart security systems too were observed to be crippled by scalability when utilized for high-speed data streams (Ramapatruni et al., 2019). The Decision Tree breaking this issue by achieving quicker inference with ability to support robustness with high-scale cloud datasets (Saad et al., 2011).

From the perspective of a cybersecurity development, this research provides an easy solution to implement on current monitoring models, real-time classification without a need for substantial computational setup (Saeedi, 2019). Other studies also noted that DDoS counter measures from blockchain technology, promising as they were, would be more likely beset with deployment complexity and hence light solutions like Decision Trees are more feasible for rapid deployment on company systems (Singh et al., 2020). Research on blockchain-based DDoS mitigation also proposed the need for its integration with active prevention based on machine learning detection, a focus where Decision Trees can strengthen distributed environments (Wani et al., 2021). Machine learning of detecting botnet traffic also demonstrated interpretability to be the essence in separating malicious activity, a lesson also found in this research (Stevanovic and Pedersen, 2016).

Finally, this study contributes to the knowledge by bridging gaps in scalability, interpretability, and flexibility, which were previously recognized as drawbacks in cloud-based detection systems (Saghezchi et al., 2022). It contributes to ongoing controversy in machine learning research where transparency and fairness in performance are key challenges in security application (Sahu et al., 2023). In addition, it has practical use in cloud-specific DDoS detection by combining dependency on binary classifiers with the incorporation of multi-class differences for enhanced detection and diminished false positives (Saini et al., 2020). On the whole, the Decision Tree Classifier is beneficial to the discipline in that it provides a robust, interpretable, and computationally lightweight model that enhances real-time protection mechanisms in clouds while providing a solid basis for future cybersecurity construction (Singh et al., 2020).

8. CONCLUSION

Cloud platforms are central to modern digital infrastructure, yet are extremely vulnerable to Distributed Denial of Service attacks due to their scalability and shared resources. Classical detection methods have fallen short, necessitating the use of adaptive machine learning methods. The study compared certain classifiers and believed the best to be the Decision Tree, with 96.8 percent accuracy, balanced precision, recall, and F1-measures of 88.9 percent and an inference time of 0.1029 seconds. This establishes its appropriateness for real-time DDoS detection in cloud environments.

The research contributes to the base of earlier work by moving away from binary detection to multi-class classification, so as to provide improved detection of malicious traffic that is often absent in earlier works. In contrast to sophisticated models that have high resource needs, Decision Tree provides performance alongside interpretability, filling a wide usability gap. This contribution empowers developers with a deployable solution alongside laying the groundwork for scalable, interpretable, and efficient cloud cybersecurity defense.

9. RESEARCH CHALLENGES

The challenges encountered in this research are mostly attributed to the intricacy of dealing with real-time actual cloud traffic data and the inability of certain machine learning algorithms to yield balanced detection for all traffic categories. Class imbalance, for example, was one of the problems encountered in the form of suspicious traffic detection, which resulted in models like Logistic Regression and Ridge Classifier performing poorly in giving consistent precision and recall.

The second issue was computational cost, because models like the K-Nearest Neighbors were very accurate, but inference times were very slow, meaning they could not be used in real-time deployment.

In addition, training models to generalize well across traffic conditions was difficult because other algorithms would break when presented with overspilled traffic patterns. Another limitation is the reliance on a single data set, which limits testing against adaptive and adaptive Distributed Denial of Service tactics. These challenges point out the trade-offs in usability on actual deployments, efficiency, and precision.

10. FUTURE WORK AND RECOMMENDATIONS

Future research will need to complement this work by testing the Decision Tree model on heterogeneous cloud deployments and various real-world datasets in order to make it more generalizable. Although the model performed extremely well in controlled environments, its performance in multi-tenant and cross-provider environments could introduce new issues necessitating further refinements. Another significant direction is to counter low-rate and stealth Distributed Denial of Service attacks, which go unnoticed with the light traffic they generate. The incorporation of advanced feature selection techniques or hybrid models that leverage Decision Trees in conjunction with ensemble learning or deep learning models would further improve detection accuracy without compromising computational performance. Furthermore, adaptive learning processes to acquire the ability to adapt traffic patterns and concept shift in cloud networks will make the system more robust. Finally, explainable artificial intelligence methods will be employed to provide greater transparency so that cybersecurity experts are able to comprehend and believe in model choices so as to properly implement them in mission-critical cloud infrastructure.

REFERENCES

[1] A. Bhardwaj, V. Mangat, and R. Vig, “Hyperband Tuned Deep Neural Network With Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud,” *IEEE Access*, vol. 8, pp. 181916–181929, 2020, doi: https://doi.org/10.1109/access.2020.3028690.

[2] M. Alduailij, Q. W. Khan, M. Tahir, M. Sardaraz, M. Alduailij, and F. Malik, “Machine-Learning-Based DDoS Attack Detection Using Mutual Information and Random Forest Feature Importance Method,” *Symmetry*, vol. 14, no. 6, p. 1095, May 2022, doi: https://doi.org/10.3390/sym14061095.

[3] M. J. Awan et al., “Real-Time DDoS Attack Detection System Using Big Data Approach,” *Sustainability*, vol. 13, no. 19, p. 10743, Sep. 2021, doi: https://doi.org/10.3390/su131910743.

[4] O. Bamasag, A. Alsaeedi, A. Munshi, D. Alghazzawi, S. Alshehri, and A. Jamjoom, “Real-time DDoS flood attack monitoring and detection (RT-AMD) model for cloud computing,” *PeerJ Comput. Sci.*, vol. 7, p. e814, Jun. 2022, doi: https://doi.org/10.7717/peerj-cs.814.

[5] A. V. Songa and G. R. Karri, “An integrated SDN framework for early detection of DDoS attacks in cloud computing,” *J. Cloud Comput.*, vol. 13, no. 1, Mar. 2024, doi: https://doi.org/10.1186/s13677-024-00625-9.

[6] I. AlSaleh, A. Al-Samawi, and L. Nissirat, “Novel Machine Learning Approach for DDoS Cloud Detection: Bayesian-Based CNN and Data Fusion Enhancements,” *Sensors*, vol. 24, no. 5, p. 1418, Feb. 2024, doi: https://doi.org/10.3390/s24051418.

[7] F. B. Saghezchi, G. Mantas, M. A. Violas, A. M. de Oliveira Duarte, and J. Rodriguez, “Machine Learning for DDoS Attack Detection in Industry 4.0 CPPSs,” *Electronics*, vol. 11, no. 4, p. 602, Feb. 2022, doi: https://doi.org/10.3390/electronics11040602.

[8] Y. Matsuzaka and Y. Uesawa, “Ensemble Learning, Deep Learning-Based and Molecular Descriptor-Based Quantitative Structure–Activity Relationships,” *Molecules*, vol. 28, no. 5, p. 2410, Mar. 2023, doi: https://doi.org/10.3390/molecules28052410.

[9] Y. Lee, B. Park, M. Jo, J. Lee, and C. Lee, “A quantitative diagnostic method of feature coordination for machine learning model with massive data from rotary machine,” *Expert Syst. Appl.*, vol. 214, p. 119117, Mar. 2023, doi: https://doi.org/10.1016/j.eswa.2022.119117.

[10] M. K. Jamieson, G. H. Govaart, and M. Pownall, “Reflexivity in Quantitative research: a Rationale and beginner’s Guide,” *Soc. Pers. Psychol. Compass*, vol. 17, no. 4, pp. 1–15, Feb. 2023, doi: https://doi.org/10.1111/spc3.12735.

[11] U. Islam et al., “Detection of Distributed Denial of Service (DDoS) Attacks in IoT Based Monitoring System of Banking Sector Using Machine Learning Models,” *Sustainability*, vol. 14, no. 14, p. 8374, Jan. 2022, doi: https://doi.org/10.3390/su14148374.

[12] K. Hendren, K. Newcomer, S. K. Pandey, M. Smith, and N. Sumner, “How Qualitative Research Methods can be Leveraged to Strengthen Mixed Methods Research in Public Policy and Public Administration,” *Public Admin. Rev.*, vol. 83, no. 3, pp. 468–485, 2023.

[13] I. H. Sarker, A. I. Khan, Y. B. Abushark, and F. Alsolami, “Internet of Things (IoT) Security Intelligence: A Comprehensive Overview, Machine Learning Solutions and Research Directions,” *Mobile Netw. Appl.*, Mar. 2022, doi: https://doi.org/10.1007/s11036-022-01937-3.

[14] A. Mihoub, O. B. Fredj, O. Cheikhrouhou, A. Derhab, and M. Krichen, "Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques," *Comput. Electr. Eng.*, vol. 98, p. 107716, Mar. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.107716.

[15] Z. Liu, L. Qian, and S. Tang, "The prediction of DDoS attack by machine learning," in *Proc. 3rd Int. Conf. Electronics and Communication; Network and Computer Technology (ECNCT 2021)*, Mar. 2022, doi: https://doi.org/10.1117/12.2628658.

[16] V. Gaur and R. Kumar, "Analysis of Machine Learning Classifiers for Early Detection of DDoS Attacks on IoT Devices," *Arab. J. Sci. Eng.*, Jul. 2021, doi: https://doi.org/10.1007/s13369-021-05947-3.

[17] B. B. Gupta, P. Chaudhary, X. Chang, and N. Nedjah, "Smart defense against distributed Denial of service attack in IoT networks using supervised learning classifiers," *Comput. Electr. Eng.*, vol. 98, p. 107726, Mar. 2022, doi: https://doi.org/10.1016/j.compeleceng.2022.107726.

[18] M. Aslam et al., "Adaptive Machine Learning Based Distributed Denial-of-Services Attacks Detection and Mitigation System for SDN-Enabled IoT," *Sensors*, vol. 22, no. 7, p. 2697, Mar. 2022, doi: https://doi.org/10.3390/s22072697.

[19] A. A. Alashhab, M. S. M. Zahid, M. A. Azim, M. Y. Daha, B. Isyaku, and S. Ali, "A Survey of Low Rate DDoS Detection Techniques Based on Machine Learning in Software-Defined Networks," *Symmetry*, vol. 14, no. 8, p. 1563, Jul. 2022, doi: https://doi.org/10.3390/sym14081563.

[20] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, "Transport and Application Layer DDoS Attacks Detection to IoT Devices by Using Machine Learning and Deep Learning Models," *Sensors*, vol. 22, no. 9, p. 3367, Apr. 2022, doi: https://doi.org/10.3390/s22093367.

[21] S. K. Sahu, A. Mokhade, and N. D. Bokde, "An Overview of Machine Learning, Deep Learning, and Reinforcement Learning-Based Techniques in Quantitative Finance: Recent Progress and Challenges," *Appl. Sci.*, vol. 13, no. 3, p. 1956, Feb. 2023, doi: https://doi.org/10.3390/app13031956.

[22] R. Sahay, G. Blanc, Z. Zhang, and H. Debar, "ArOMA: An SDN based autonomic DDoS mitigation framework," *Comput. Secur.*, vol. 70, pp. 482–499, Sep. 2017, doi: https://doi.org/10.1016/j.cose.2017.07.008.

[23] S. Sadhwani, B. Manibalan, R. Muthalagu, and P. M. Pawar, "A Lightweight Model for DDoS Attack Detection Using Machine Learning Techniques," *Appl. Sci.*, vol. 13, no. 17, p. 9937, Sep. 2023, doi: https://doi.org/10.3390/app13179937.

[24] T. E. Ali, Y.-W. Chong, and S. Manickam, "Comparison of ML/DL Approaches for Detecting DDoS Attacks in SDN," *Appl. Sci.*, vol. 13, no. 5, p. 3033, Feb. 2023, doi: https://doi.org/10.3390/app13053033.

[25] J. Wang and L. Wang, "SDN-Defend: A Lightweight Online Attack Detection and Mitigation System for DDoS Attacks in SDN," *Sensors*, vol. 22, no. 21, p. 8287, Oct. 2022, doi: https://doi.org/10.3390/s22218287.

[26] H. Wang and W. Li, "DDosTC: A Transformer-Based Network Attack Detection Hybrid Mechanism in SDN," *Sensors*, vol. 21, no. 15, p. 5047, Jan. 2021, doi: https://doi.org/10.3390/s21155047.

[27] Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks," *Sensors*, vol. 23, no. 13, p. 6176, Jan. 2023, doi: https://doi.org/10.3390/s23136176.

[28] S. Ali, Y. Li, and M. Uzair, "DDoS attack detection in smart grid network using reconstructive machine learning models," *PeerJ Comput. Sci.*, vol. 10, p. e1784, Jan. 2024, doi: https://doi.org/10.7717/peerj-cs.1784.

[29] S. Abbas et al., "Evaluating deep learning variants for cyber-attacks detection and multi-class classification in IoT networks," *PeerJ Comput. Sci.*, vol. 10, p. e1793, Jan. 2024, doi: https://doi.org/10.7717/peerj-cs.1793.

[30] A. K. Mousa and M. N. Abdullah, "An Improved Deep Learning Model for DDoS Detection Based on Hybrid Stacked Autoencoder and Checkpoint Network," *Future Internet*, vol. 15, no. 8, p. 278, Aug. 2023, doi: https://doi.org/10.3390/fi15080278.

[31] S. Ahmed et al., "Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron," *Future Internet*, vol. 15, no. 2, p. 76, Feb. 2023, doi: https://doi.org/10.3390/fi15020076.

[32] A. A. Alahmadi et al., "DDoS Attack Detection in IoT-Based Networks Using Machine Learning Models: A Survey and Research Directions," *Electronics*, vol. 12, no. 14, p. 3103, Jan. 2023, doi: https://doi.org/10.3390/electronics12143103.

[33] M. Shafi, A. H. Lashkari, V. Rodriguez, and R. Nevo, "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS) Dataset and Cloud Intrusion Traffic Characterization," *Information*, vol. 15, no. 4, p. 195, Apr. 2024, doi: https://doi.org/10.3390/info15040195.

[34] Y. Fu, X. Duan, K. Wang, and B. Li, "Low-rate Denial of Service attack detection method based on time-frequency characteristics," *J. Cloud Comput.*, vol. 11, no. 1, Aug. 2022, doi: https://doi.org/10.1186/s13677-022-00308-3.

[35] M. H. Mutar, A. Hani, A. Nasser, and A. Mansour, "Predicting the Impact of Distributed Denial of Service (DDoS) Attacks in Long-Term Evolution for Machine (LTE-M) Networks Using a Continuous-Time Markov Chain (CTMC) Model," *Electronics*, vol. 13, no. 21, p. 4145, Oct. 2024, doi: https://doi.org/10.3390/electronics13214145.

[36] S. Oyucu, O. Polat, M. Türkoğlu, H. Polat, A. Aksöz, and M. T. Ağdaş, "Ensemble Learning Framework for DDoS Detection in SDN-Based SCADA Systems," *Sensors*, vol. 24, no. 1, p. 155, Dec. 2023, doi: https://doi.org/10.3390/s24010155.

[37] M. Ramzan et al., "Distributed Denial of Service Attack Detection in Network Traffic Using Deep Learning Algorithm," *Sensors*, vol. 23, no. 20, p. 8642, Oct. 2023, doi: https://doi.org/10.3390/s23208642.

[38] D. Alghazzawi, O. Bamasag, H. Ullah, and M. Z. Asghar, "Efficient Detection of DDoS Attacks Using a Hybrid Deep Learning Model with Improved Feature Selection," *Appl. Sci.*, vol. 11, no. 24, p. 11634, Dec. 2021, doi: https://doi.org/10.3390/app112411634.

[39] R. Ma, X. Chen, and R. Zhai, "A DDoS Attack Detection Method Based on Natural Selection of Features and Models," *Electronics*, vol. 12, no. 4, p. 1059, Feb. 2023, doi: https://doi.org/10.3390/electronics12041059.

[40] S.-H. Lee, Y.-L. Shiue, C.-H. Cheng, Y.-H. Li, and Y.-F. Huang, "Detection and Prevention of DDoS Attacks on the IoT," *Appl. Sci.*, vol. 12, no. 23, p. 12407, Dec. 2022, doi: https://doi.org/10.3390/app122312407.

[41] Z. Long, H. Yan, G. Shen, X. Zhang, H. He, and L. Cheng, "A Transformer-based network intrusion detection approach for cloud security," *J. Cloud Comput.*, vol. 13, no. 1, Jan. 2024, doi: https://doi.org/10.1186/s13677-023-00574-9.

[42] Z. R. Alashhab, M. Anbar, M. M. Singh, I. H. Hasbullah, P. Jain, and T. A. Al-Amiedy, "Distributed Denial of Service Attacks against Cloud Computing Environment: Survey, Issues, Challenges and Coherent Taxonomy," *Appl. Sci.*, vol. 12, no. 23, p. 12441, Jan. 2022, doi: https://doi.org/10.3390/app122312441.

[43] L. Liu, W. Yu, Z. Wu, and S. Peng, "XGBoost-Based Detection of DDoS Attacks in Named Data Networking," *Future Internet*, vol. 17, no. 5, p. 206, May 2025, doi: https://doi.org/10.3390/fi17050206.

[44] J. D. Gadze, A. A. Bamfo-Asante, J. O. Agyemang, H. Nunoo-Mensah, and K. A.-B. Opare, "An Investigation into the Application of Deep Learning in the Detection and Mitigation of DDOS Attack on SDN Controllers," *Technologies*, vol. 9, no. 1, p. 14, Feb. 2021, doi: https://doi.org/10.3390/technologies9010014.

[45] D. Han, H. Li, and X. Fu, "Reflective Distributed Denial of Service Detection: A Novel Model Utilizing Binary Particle Swarm Optimization—Simulated Annealing for Feature Selection and Gray Wolf Optimization-Optimized LightGBM Algorithm," *Sensors*, vol. 24, no. 19, p. 6179, Sep. 2024, doi: https://doi.org/10.3390/s24196179.

[46] U. B. Clinton, N. Hoque, and K. R. Singh, "Classification of DDoS attack traffic on SDN network environment using deep learning," *Cybersecurity*, vol. 7, no. 1, Aug. 2024, doi: https://doi.org/10.1186/s42400-024-00219-7

.