SSRG International Journal of Electronics and Communication Engineering
ISSN: 2348-8549/ https://doi.org/10.14445/23488549/IJECE-V13I4P124

Volume 13 Issue 4, 285-296, April 2026

Original Article

Scalable and Automated Detection of Cloud-Based DDoS Attacks Using AutoML-Random Forests

Rachna Singh[1], Nitasha Soni[2]

[1,2]*Department of CSE, SET, M R I I R S, Faridabad, India.*

[1]*Corresponding Author : rachna.singh.080@gmail.com*

Received: 20 January 2026 Revised: 20 February 2026 Accepted: 22 March 2026 Published: 30 April 2026

Abstract *- With the advent of the digital era, the prevalence of cloud computing has increased, leading to a significant rise in cyber threats, specifically Denial of Service attacks. To prevent the cloud network from such attacks, several detection methods were used, namely, rule-based and signature-based, which struggle to adapt to the intricate and dynamic nature of the networks. A unique Auto-ML-based Random Forest technique is suggested to categorize cloud-based DDoS assaults in order to overcome the shortcomings of conventional approaches. A novel, BCCC-cPacket-Cloud DDoS-2024 dataset is employed, which captures diverse types of DDoS attacks. The approach automated the feature selection process, and the model is optimized to enhance its performance. In this, the top 40 most relevant features were extracted using feature reduction techniques. Further, multiple classification tasks, such as binary, ternary, and activity-specific, are focused on attaining accurate and optimized results. In comparison to state-of-the-art methods, it was shown that an accuracy of around 98% was achieved in a number of categories. Consequently, confirming the efficacy of the suggested strategy in a cloud environment is necessary.*

Keywords *- DDoS attack, Cloud security, Machine Learning, AutoML.*

1. Introduction

The advent and adoption of cloud computing have transformed the digital world, enabling individuals and businesses to access computing resources with flexibility at optimum costs. As cloud resources have become an integral part of large-scale commercial companies and personal applications, this leads to critical security concerns worldwide [1]-[4]. One of the most frequent and serious assaults that impairs network functioning and resources and causes harm is Distributed Denial of Service (DDoS) attacks. With the progression in technological advancements, these attacks have also evolved in terms of complexity and several types, namely, protocol, volumetric, and application. Therefore, when traditional DDoS attacks were detected by traditional methods such as signature-based or rule-based methods, these methods now struggle to detect and prevent evolved DDoS attacks due to a lack of adaptability to dynamic patterns. Also, the massive and varied volume of traffic generated by cloud services makes them inefficient and results in delayed response times [5]-[6]. This leads to the growing need for novel, automated, and scalable solutions to detect evolved DDoS attacks that can also handle the complexities of cloud networks [7]-[9].

Existing approaches rely on various machine learning models such as Decision Trees, SVM, (KNN), or deep learning methods. Although these models achieve good results, they frequently require manual hyperparameter tuning and sometimes suffer from overfitting when high-dimensional network features are used. Moreover, prior studies do not systematically analyze feature importance or validate feature reduction strategies in cloud-specific datasets.

Although many studies have applied machine learning and deep learning for DDoS detection, some important gaps still remain:

1. Outdated datasets – Many works use older or generic datasets that do not reflect today's dynamic cloud traffic patterns.
2. Only binary focus – Most research limits detection to benign vs. attack, ignoring suspicious or intermediate traffic behavior.
3. Limited automation – Automated model optimization using AutoML is rarely explored in cloud-based DDoS detection.
4. Less attention to feature reduction – High-dimensional traffic data is often used without properly selecting the most relevant features.
5. Lack of fine-grained classification – Few studies differentiate specific attack types (e.g., TCP SYN) from normal activities like SSH or HTTP-S.

These gaps highlight the need for a more practical, optimized, and detailed classification approach for cloud environments. To overcome these challenges of traditional methods and to detect and classify evolved cloud-based DDoS attacks, a novel approach is proposed based on AutoML and

the random Forest algorithm. In this approach, automated feature selection is integrated with model optimization. Additionally, cross-validation is used to lower computational overhead and improve detection efficiency and accuracy. The suggested method is tested on extensive and recently created datasets [10] that represent current patterns in cloud traffic, offering a strong basis for model evaluation and training.

Major contributions of this work are:

- To determine the top 40 most pertinent characteristics, a feature reduction strategy that combines principal component analysis and correlation analysis is used.
- Applying a number of algorithms, including support vector machines, random forests, and decision trees, along with H2O AutoML for hyperparameter optimization, it achieved a 97% accuracy rate across several DDoS attacks.

2. Related Work

The detection and prevention of DDoS attacks has been a significant research topic, with many different approaches suggested. This section provides a brief overview of some of the various approaches proposed for detecting DDoS attacks.

2.1. Related Work on DDoS Attacks in Cloud Environment

Signature-based methods were initially proposed to detect DDoS attacks. Peng et al. proposed an approach to recognize previously recorded attacks, but were unable to detect new or unknown patterns [11]. Xu et al. proposed anomaly-based systems to predict these attacks in cloud systems [12]. Several approaches were proposed to detect DDoS attacks in cloud systems, such as access restrictions, resource allocation limitations, and others. To mitigate DDoS attacks in cloud environments, various techniques have been introduced, such as challenge-response methods, unseen server configurations, access restrictions, and resource allocation limitations. Puzzle-based approaches, like CAPTCHA, have also been employed as a straightforward defense mechanism, but recent studies have demonstrated their limited effectiveness. Another approach, the IP traceback defense mechanism, combines packet marking, reconstruction, and entropy-based techniques to identify the source of attacks. Dantas et al. [13] proposed an Adaptive Selective Verification-based approach to prevent network-layer DDoS attacks.

2.2. Related work on Machine Learning-based DDoS Detection

Several machine learning approaches have been proposed to detect and classify DDoS attacks. Alkasassbeh et al. [14] made use of contemporary DDoS attack data, which had been collected from a variety of network layers, such as SIDDoS and HTTP flood attacks. The authors were able to compare 27 different features to examine various machine learning techniques. In the study by Carl Livadas et al. [15], various machine learning techniques were deployed to identify malicious traffic from Command and Control (C&C) servers that may perform DDoS attacks, with results demonstrating that Naïve Bayes performed better than other techniques. A review of the literature by Bayu Adhi Tama et al. [16] studied 35 papers on DDoS and DoS attacks, specifically in terms of data mining methods. Several machine learning models, supervised, unsupervised, and hybrid, have been used to address the DDoS detection issue.

In their study, Liao et al. [17] developed a detection mechanism that employs a Support Vector Machine (SVM). It recorded user request patterns via parameters such as request frequency sequence, benefiting from the fact that bots tend to visit web pages in a similar manner. Thus, compared patterns were determined by matching rhythm. Xiao et al. [18] proposed another method, using K Nearest Neighbors (KNN) to derive a detection method based on flows generated by the same program, affecting correlation. An anomaly-based DDoS detection approach considering packet features was proposed in [19], applying Radial Basis Function neural networks. Similarly, [20] FCM clustering and a priori association algorithms were deployed to extract models of network traffic and protocol status, regarding a threshold for detection. Also, in [21], a detection approach using a decision tree and grey relational analysis was proposed. Jie-Hao et al. [22] applied an Artificial Neural Network (ANN) for DDoS attack detection, comparing performance to decision trees, entropy, and Bayesian. Further, Liu et al. [23] deployed a Learning Vector Quantization neural network method for pattern recognition and compression. In [24], to classify DDoS attacks, a Probabilistic Neural Network-based approach was proposed. A novel approach for intrusion prevention was proposed by Li et al. [25], which integrates SNORT and SVM, to reduce false alarm rates. A Distributed Time Delay Neural Network [26] attains higher accuracy for network attacks. Further, protocol analysis and clustering were used to classify DDoS attacks. [27] has been extensively used in DDoS attack detection.

In this work, the BCCC-cPacket-Cloud DDoS-2024 dataset has been leveraged to bridge the gaps to take a more thorough approach. Comparative analysis of the state-of-the-art methods is listed in Table 1.

Many academics have already presented their work and offered fresh, perceptive information on DDOS assaults; nevertheless, no comprehensive approach has been devised to create a system that can recognize the many DDOS attack types in cloud environments. In the suggested way, we have classified all of the attack information using the AutoML method along with the Random Forest machine learning model. First, the three main classes, benign, assault, and suspicious, have been determined. Following that, we also had to categorize the various signature-based techniques. The proposed method accurately and efficiently detects different types of DDOS attacks in the cloud environment and

outperforms the existing significant methods. The proposed research used a novel and optimized framework for cloud-based DDoS and intrusion traffic classification using AutoML with Random Forest. The in-between process is also combined with feature reduction techniques using PCA and correlation analysis. The novelty of this work lies in:

1. The proposed framework provides behavioral traffic analysis specifically tailored for cloud environments**,** where packet inter-arrival time patterns and payload variations are analyzed to capture subtle differences between benign, suspicious, and attack traffic. This enables a deeper understanding of traffic behavior rather than simple attack detection.
2. The proposed framework integrates AutoML-driven optimization with structured feature reduction**,** allowing the model to automatically identify the most important traffic characteristics, minimizing redundancy. This improves detection efficiency and model acceptability, which is often lacking in deep learning–based approaches.
3. A structured feature reduction process to identify the top 50 most discriminative network features.
4. Integration of AutoML for automated hyperparameter tuning and model optimization.
5. Multi-level classification, including:
   - Binary classification (Benign vs. Attack),
   - Ternary classification (Benign, Attack, Suspicious),
   - Fine-grained activity-level categorization (e.g., Attack-TCP-Valid-SYN, SSH, FTP, HTTP-S, Email traffic).

Unlike traditional approaches that focus on detection, the proposed framework provides deeper traffic behavior analysis, making it more practical for real-world cloud security applications.

3. Research Methodology

In this section, the dataset, the applied machine learning techniques, and the proposed methodology, including pseudocode, are described.

- Datasets
- Feature Extraction
- Proposed Methodology
- Proposed pseudocode
- Model Evaluation
- Implementation

Table 1. Comparative analysis of state-of-the-art approaches

Study	Techniques/Methodologies	Datasets	Evaluation Metrics	Attack Types Detected
Alkasassbeh et al. [14]	Multilayer perceptron (MLP), Random Forest, Naïve Bayes	SIDDoS, HTTP flood	Accuracy, Precision, Recall	Volumetric DDoS
Livadas et al. [15]	Naïve Bayes, J48, Bayesian Network	Command and control (C&C) traffic	Detection Rate, False Positive Rate	IRC-based DDoS
Tama et al. [16]	Literature review (35 papers)	Multiple Datasets	N/A (survey paper)	Various DDoS
Liao et al. [17]	Support Vector Machine (SVM) Rhythm-Matching Algorithm	N/A	Precision, Recall, F1-score	Application Layer DDoS
Xiao et al. [18]	k-Nearest Neighbors (KNN)	N/A	Detection Accuracy, False Positive Rate	Botnet-based DDoS
Karimazad et al. [19]	Radial Basis Function (RBF) Neural Networks	N/A	Classification Accuracy, False Positive Rate	Packet-based DDoS
Zhong et al. [20]	FCM Clustering, A priori Association Algorithm	N/A	Detection Rate, False Negative Rate	Protocol-based DDoS
Wu et al. [21]	Decision Tree, Grey Relational Analysis	N/A	Classification Accuracy, Detection Rate	TCP-based DDoS
Chen et al. [24]	Artificial Neural Networks (ANN), Decision Trees, Entropy, Bayesian	N/A	Detection Accuracy, Precision, Recall	Application Layer DDoS
Liu et al. [22]	Learning Vector Quantization (LVQ) Neural Networks	N/A	Classification Accuracy,	Volumetric & Application DDoS

			Precision, Recall	
Akilandeswari et al. [23]	Probabilistic Neural Networks (PNN), Bayes Decision Rule, Radial Basis Function Neural Networks (RBFNN)	N/A	Classification Accuracy, False Positive Rate	Flash Crowd vs. DDoS
Li et al. [25]	Support Vector Machine (SVM), SNORT, Configurable Firewall	N/A	False Alarm Rate, Accuracy	Intrusion Detection & DDoS
Ibrahim et al. [26]	Distributed Time Delay Neural Networks (DTDNN)	N/A	Detection Accuracy, Conversion Rate	General Network Attacks

3.1. Datasets

In order to record cloud-based Distributed Denial of Service (DDoS) assaults and cloud infiltration traffic patterns, this study used a recently created dataset. The 2024-generated dataset reflects current traffic patterns and threats and is a useful tool for enhancing cloud computing DDoS detection and mitigation techniques. Table 2 provides an explanation of key aspects.

Table 2. Key Features of the dataset

Feature	Description
Traffic Types	It includes several attacks, such as volumetric, protocol-based, and application-layer-based.
Intrusion Traffic	Logs intrusion attempts targeting cloud services, providing insights into malicious activities beyond DDoS attacks.
Temporal Data	Contains timestamps for each traffic sample, allowing analysis of attack patterns over time and study of DDoS attack dynamics.
Traffic Characteristics	Provide details such as packet size, source and destination IP addresses, protocol types, and traffic volume, essential for analyzing and modelling network behaviour.
Cloud-Specific Attributes	Focuses on cloud traffic features like scalability, multi-tenancy, and virtualized network elements, specific to cloud environments.
Diverse Attack Vectors	Encompasses a variety of attack vectors, enabling researchers to study different attacker strategies and assess the effectiveness of defence mechanisms.
Anomaly Labels	Each data entry is labeled as “attack” or “normal,” supporting supervised learning methods for model training and evaluation.
Synthetic Data Generation	Includes synthetic traffic generated under controlled conditions, simulating realistic DDoS and intrusion scenarios to represent a wide range of attack types.

3.2. Feature Extraction

The visualization of the BCCC-cPacket-Cloud-DDoS-2024 dataset covers over 300 features from the network and transport layers, collected via a tool called NTLFlowLyzer [28-29]. Among these features, the top 40 features were extracted using an accepted best feature selector method. The features extracted are presented in Table 3.

3.3. Proposed Methodology

Figure 1 shows the proposed method to detect and classify DDoS attacks in a cloud environment. In this approach, an AutoML-optimized Random Forest model is integrated with feature reduction methods. This approach is specifically based to detect DDoS attacks in a cloud environment. Firstly, the top 40 relevant features were extracted, after which the model was trained for binary (benign and attack) and ternary (benign, attack, and suspicious) classification. The model was further refined to categorize the data into specific activities, namely, Attack-TCP-Valid-SYN, SSH activities, and others. The choice of Random Forest was motivated by the characteristics of cloud network traffic data, which contain structured statistical features rather than raw sequential or image-like inputs typically required for deep learning models.

Random Forest effectively captures nonlinear relationships among traffic features while maintaining high interpretability and lower computational overhead. When combined with AutoML, the framework automatically optimizes hyperparameters and model configurations, enabling performance comparable to or exceeding deep learning approaches without requiring extensive training resources.

Furthermore, deep learning models such as Graph Neural Networks and GAN-based frameworks often demand high computational cost, large-scale labeled datasets, and complex tuning processes. In contrast, the proposed approach achieves near state-of-the-art detection performance while remaining lightweight, scalable, and suitable for real-time deployment in cloud environments.

Table 3. Top 40 selected features

Activity, fwd_packets_IAT_mode, packet_IAT_max, fwd_packets_IAT_mean, max_packets_delta_len, fwd_packets_IAT_median, fwd_packets_IAT_min, fwd_packets_IAT_total, packets_IAT_mode, fwd_packets_IAT_max, min_bwd_packets_delta_time, max_bwd_packets_delta_len, packets_IAT_mean, mode_bwd_packets_delta_time, mean_bwd_packets_delta_time, packets_IAT_median, median_bwd_packets_delta_time, median_packets_delta_time, bwd_packets_count, bwd_packets_IAT_median, packet_IAT_total, skewness_packets_delta_time, bwd_payload_bytes_max, mean_packets_delta_time, median_bwd_packets_delta_len, packet_IAT_min, max_bwd_payload_bytes_delta_len, fwd_payload_bytes_max, psh_flag_percentage_in_total, bwd_ack_flag_counts, mode_packets_delta_time, bwd_packets_IAT_mode, bwd_init_win_bytes, bwd_packets_IAT_max, max_bwd_packets_delta_time, bwd_packets_IAT_mean, mean_fwd_packets_delta_len, fwd_init_win_bytes, std_packets_delta_time, payload_bytes_max

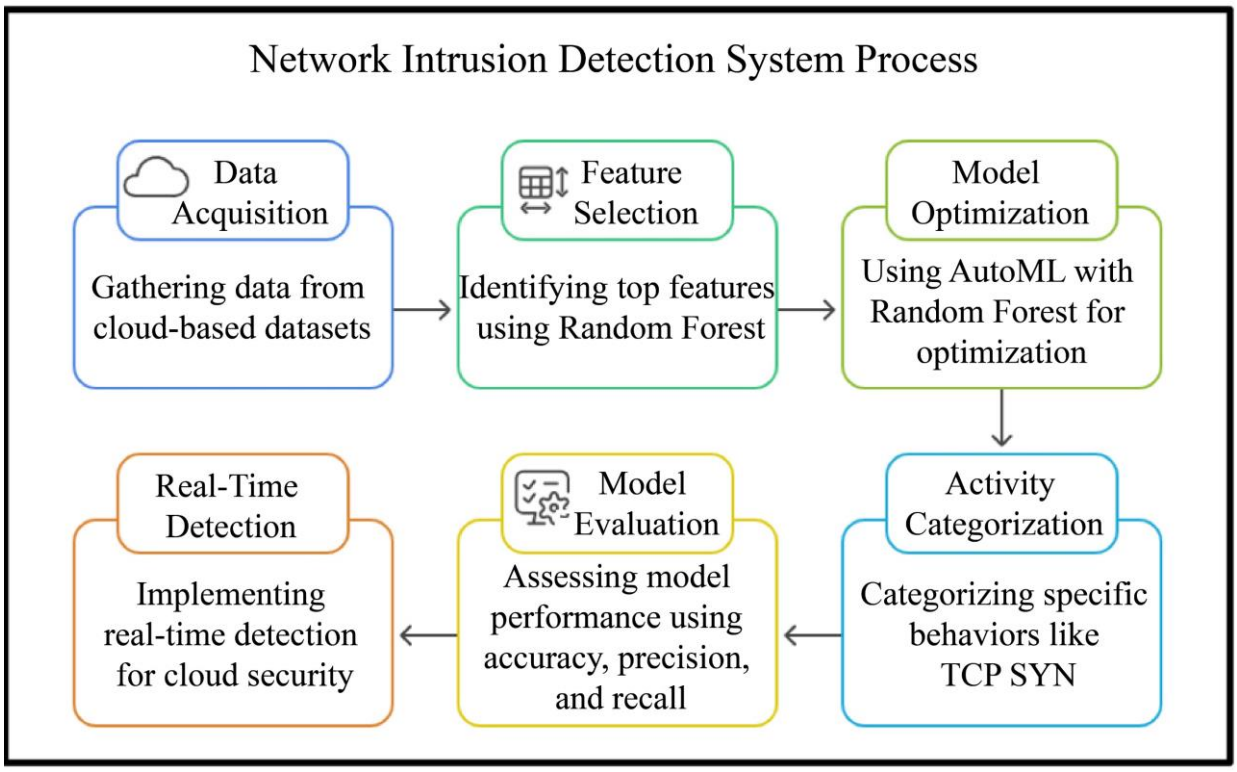


Fig. 1 Proposed Methodology

The proposed model was able to distinguish between the complicated traffic patterns, and the performance evaluation was reported on multiple performance measures, including precision, recall, and accuracy. The model optimization process has been automated using AutoML, and when integrated with random forest and feature reduction techniques, it was found to be highly effective in classifying DDoS attacks and intrusion traffic.

3.4. Proposed Pseudocode

This proposed pseudocode presents an AutoML-driven method. The dataset was first pre-processed by eliminating missing values and redundant or inconsistent data. Let the dataset be represented as;

$$D = \{X,Y\}$$

where $X=\{x_1,x_2,...,x_n\}$denotes the original feature space, and *Y* represents class labels.

A Random Forest model computes feature importance using impurity reduction:

$$FI_j = \sum_{t=1}^{T} \Delta I_{j,t}$$

where is the importance of feature *J*, *T* is the number of trees, and $\Delta I_{j,t}$represents the decrease in node impurity contributed by feature *j* in tree t.

Features are ranked according to normalized importance scores: The top *K* features are selected:

X_selected=Top K(FI', K)

The reduced dataset becomes:

D_reduced={X_selected,Y}

Thus, the final feature set consists exclusively of selected original variables, not transformed dimensions. The proposed framework applies Feature Selection, not Feature Extraction. The top k characteristics were found following pre-processing. The top 40 features were chosen after feature reduction techniques were used to determine which of these k characteristics were the most important.

By doing this, the data's dimensionality is decreased, increasing the model's effectiveness. After the feature extraction, we partitioned the dataset into training and testing datasets and applied an AutoML model with a random forest.

Further, the model was cross-validated with different numbers of folds to attain the most accurate results. The model classified the datasets into binary (attack vs. no attack) and ternary (e.g., normal, DDoS, and intrusion). After each iteration, the performance of the model was evaluated, and the best-performing model was attained.

3.5. Model Evaluation

The proposed approach is evaluated using various performance metrics as presented.

$$\text{Precision} = \frac{T_P}{T_P + F_P} \quad (1)$$

$$\text{Recall} = \frac{T_P}{T_P + F_N} \quad (2)$$

$$\text{Accuracy} = \frac{T_P + T_N}{N} \quad (3)$$

$$\text{F1-Score} = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

3.6. Implementation

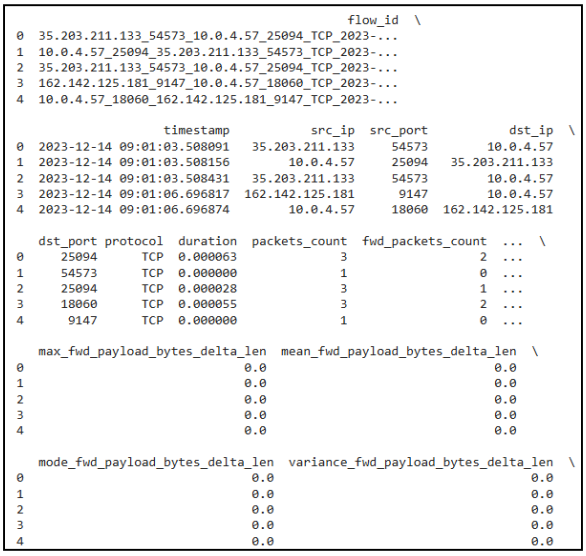
In this section, we demonstrate the proposed methodology in a step-by-step and applied manner. Data preprocessing, model development, system architecture, and the deployment environment were covered. The implemented code is designed in Python, and the Google colab environment is used to run the programming part.

3.6.1. Data Pre-processing and Feature Selection

This work utilized a large dataset tailored for intrusion traffic characterization in the cloud as well as Distributed Denial of Service (DDoS) attacks. The dataset underwent pre-processing to remove incomplete or inconsistent data to ensure high-quality data integrity. Figure 2 shows the file reading and the first five rows of the data in the Python environment.

Algorithm: Cloud-based DDoS and Intrusion Traffic Classification using AutoML with Random Forest

Input:
D := {$x_1, x_2, ..., x_n$} // Original dataset, K ∈ ℕ // Number of top features
R ∈ [0,1] // Training split ratio, F ∈ ℕ // Number of cross-validation folds
Output:
M* // Optimal model, P // Performance metrics
Begin:
// Data preprocessing
D_clean ← RemoveNoise(D)
F_top ← {$f_1, f_2, ..., f_k$} where f_i ∈ TopFeatures(D_clean, K)
D_reduced ← π_F_top(D_clean)
// Project data onto selected features
// Class labeling
for each type ∈ {binary, ternary} do
D[type] ← AssignLabels(D_reduced, type)
end for
// Dataset splitting
D_train ← {x ∈ D_reduced | index(x) ≤ ⌊|D_reduced| × R⌋}
D_test ← D_reduced \ D_train
// Initialize AutoML
M* ← ∅ // Best model
P* ← -∞ // Best performance
RF_params ← RandomForestParameters()
// Cross-validation training
for i ← 1 to F do
M_i ← AutoML_Train(D_train, RF_params)
P_i ← Evaluate(M_i, D_test)
if P_i > P* then
M* ← M_i
P* ← P_i
end if
end for
// Multi-class evaluation
for each type ∈ {binary, ternary} do
P[type] ← Evaluate(M*, D[type])
CM[type] ← ComputeConfusionMatrix(M*, D[type])
end for
// Performance metrics calculation
Function Evaluate(M, D):
return {
Accuracy = (TP + TN)/(TP + TN + FP + FN),
Precision = TP/(TP + FP),
Recall = TP/(TP + FN)
}



```
                                              flow_id  \
0  35.203.211.133_54573_10.0.4.57_25094_TCP_2023-...
1  10.0.4.57_25094_35.203.211.133_54573_TCP_2023-...
2  35.203.211.133_54573_10.0.4.57_25094_TCP_2023-...
3  162.142.125.181_9147_10.0.4.57_18060_TCP_2023-...
4  10.0.4.57_18060_162.142.125.181_9147_TCP_2023-...

                  timestamp          src_ip  src_port          dst_ip  \
0  2023-12-14 09:01:03.508091   35.203.211.133     54573        10.0.4.57
1  2023-12-14 09:01:03.508156        10.0.4.57     25094   35.203.211.133
2  2023-12-14 09:01:03.508431   35.203.211.133     54573        10.0.4.57
3  2023-12-14 09:01:06.696817  162.142.125.181      9147        10.0.4.57
4  2023-12-14 09:01:06.696874        10.0.4.57     18060  162.142.125.181

   dst_port protocol  duration  packets_count  fwd_packets_count  ...  \
0     25094      TCP  0.000063              3                  2  ...
1     54573      TCP  0.000000              1                  0  ...
2     25094      TCP  0.000028              3                  1  ...
3     18060      TCP  0.000055              3                  2  ...
4      9147      TCP  0.000000              1                  0  ...

   max_fwd_payload_bytes_delta_len  mean_fwd_payload_bytes_delta_len  \
0                              0.0                               0.0
1                              0.0                               0.0
2                              0.0                               0.0
3                              0.0                               0.0
4                              0.0                               0.0

   mode_fwd_payload_bytes_delta_len  variance_fwd_payload_bytes_delta_len  \
0                               0.0                                  0.0
1                               0.0                                  0.0
2                               0.0                                  0.0
3                               0.0                                  0.0
4                               0.0                                  0.0
```

Fig. 2 Reading the file in the Python environment

Some irrelevant features are removed manually as depicted in Figure 3.

```
# Dropping irrelevant columns
my_df = my_df.drop(["flow_id", "timestamp", "src_ip", "dst_ip", "dst_port", "cov_fwd_payload_bytes_delta_len"],
                   axis=1, errors='ignore')
```

Fig. 3 Irrelevant feature removal

Then, the top 40 features were identified based on their relevance and contribution to the classification task, based on statistical measures such as correlation and principal component analysis. Figure 4 shows the relevance of the features as per importance.

feature	importance
src_port	0.001939
protocol	0.000000
duration	0.003319
packets_count	0.001459
fwd_packets_count	0.000683
...	...
variance_fwd_payload_bytes_delta_len	0.000811
std_fwd_payload_bytes_delta_len	0.000651
median_fwd_payload_bytes_delta_len	0.000352
skewness_fwd_payload_bytes_delta_len	0.000694
activity	0.253819

[317 rows x 2 columns]

Fig. 4 Relevance score of each feature

This was a vital component of differentiating between benign, attack, and suspicious traffic. Key metrics, including inter-arrival times, packet sizes, and payload characteristics, were among the selected features. These features, which are detailed in Table 3, facilitated a focused analysis and significantly enhanced the classification accuracy of the system. Then, a refined dataset was generated, enhancing model efficiency and focus by eliminating irrelevant data points. The data is big in size, so only 30000 samples have been used for the training and testing phase, as depicted in Figure 5.

```
# Splitting Attack, Benign, and Suspicious Data
df_attack = df[df['label'] == "Attack"][:10000]  # Limiting attack data to 10,000 samples
df_benign = df[df['label'] == "Benign"][:10000]  # Limiting benign data to 10,000 samples
df_suspicious = df[df['label'] == "Suspicious"][:10000]  # Limiting suspicious data to 10,000 samples
```

Fig. 5 Sample selection

This dataset was then partitioned into training and testing datasets based on a certain split (80/20). Subsequently, the AutoML Framework was launched in Python using the h20 library.

For further feature reduction and faster processing, we then used the H2O Python module. Some of the features of h20 is shown in Figure 6.

Random Forest was then selected as the primary classification algorithm. Using the selected features, multiple models were automatically trained on the training set. To ensure robustness and avoid overwriting, a 5-fold cross-validation technique was adopted. Then, Model Optimization was performed using AutoML with Random Forest through automated hyperparameter tuning. Finally, the dataset was classified into binary and ternary classification.

H2O_cluster_uptime:	07 secs
H2O_cluster_timezone:	Etc/UTC
H2O_data_parsing_timezone:	UTC
H2O_cluster_version:	3.46.0.1
H2O_cluster_version_age:	26 days
H2O_cluster_name:	H2O_from_python_unknownUser_pxibl9
H2O_cluster_total_nodes:	1
H2O_cluster_free_memory:	3.170Gb

Fig. 6 H20 Features Set

4. Results and Discussions

This part presents a thorough study and assessment of the suggested technique using the chosen dataset, based on the several performance measures previously mentioned.

- Binary Classification
- Ternary Classification
- Multi-Activity Classification

4.1. Binary Classification

The suggested model had a 98.7% accuracy rate when used to classify traffic into two categories: benign and assault. When compared to a number of machine learning techniques, the suggested model's performance significantly improves. Table 4 and Figure 7 present the findings.

4.2. Ternary Classification

The proposed model was used for ternary classification among benign, attack, and suspicious traffic and demonstrated the best performance. The result highlights that the model is able to handle multi-class categorization with precision. The outcomes are illustrated in Figure 8.

Table 4. Results of Binary Classification

Model Applied	Accuracy	F1 Score	Precision	Recall
Logistic Regression	0.8031	0.7938	0.7945	0.8031
Decision Tree	0.949	0.9386	0.9386	0.939
Random Forest	0.9528	0.9524	0.9533	0.9528
Gradient Boost	0.9586	0.9576	0.9566	0.9556
Ada boost	0.5104	0.487	0.4408	0.6104
Proposed Approach	0.987	0.989	0.9891	0.97495

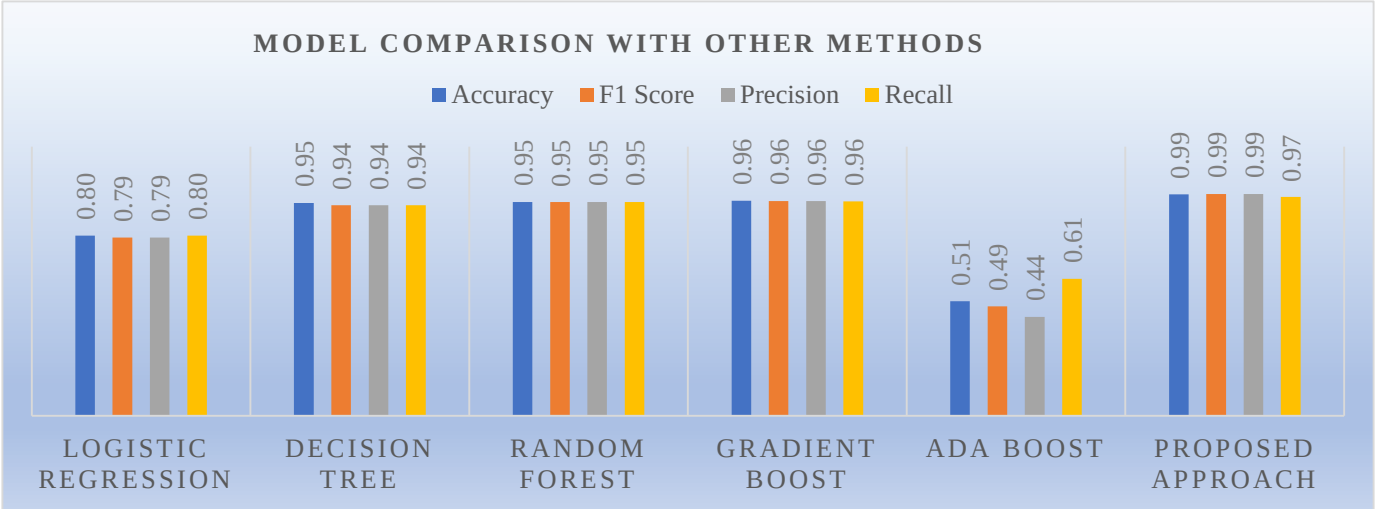


Fig. 7 Model Comparison with Other Methods

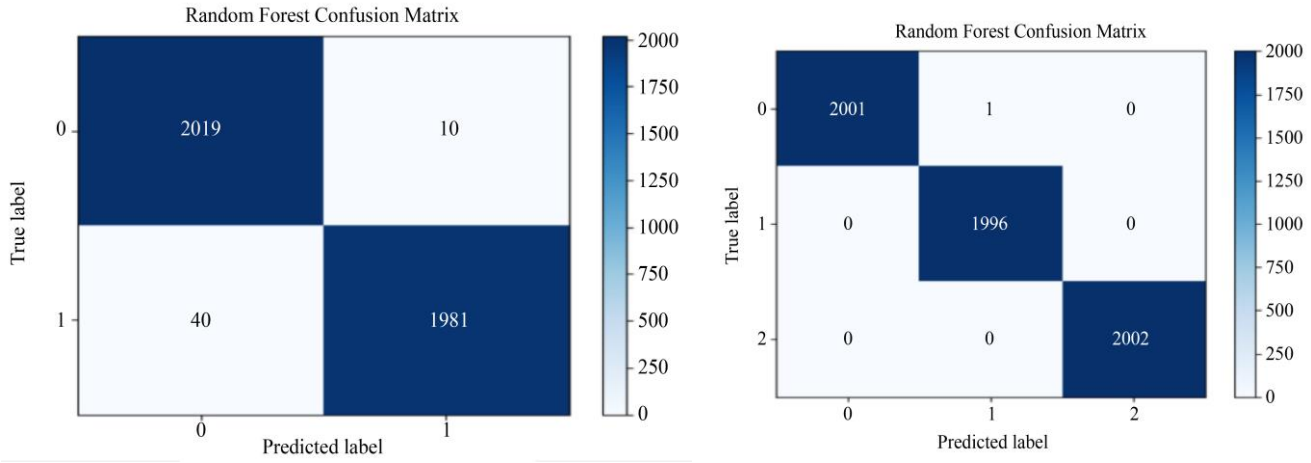


Fig. 8 Results of Binary Classification

4.3. Multi-Activity Classification

The proposed model was further extended to classify the data into multiple specific activities, such as:

- Suspicious: 10,000 instances
- Attack-TCP-Valid-SYN: 8,043 instances
- Benign activities, including: 5738 instances
- Web Browsing HTTP-S (2,504 instances)
- SSH (895 instances)
- Telnet (535 instances)
- Others, such as Email-Send, Email-Receive, and FTP (289 instances)

This further classification enhanced the granularity of the model, reinforcing its adaptability in recognizing and differentiating among a wide range of traffic types. An accuracy of 78.9% is attained for multiclass classification. The confusion matrix of multiclass classification is depicted in Figure 9.

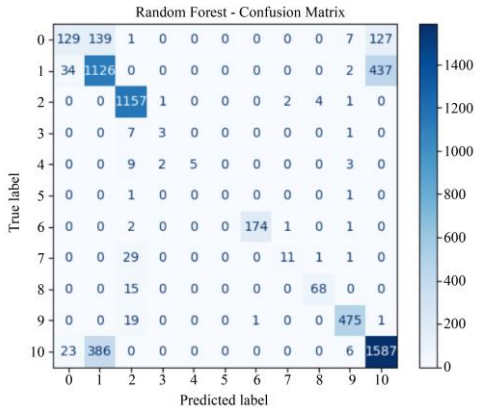


Fig. 9 Results of *Ternary* Classification

Unlike traditional approaches that focus on detection, the proposed framework provides deeper traffic behavior analysis, making it more practical for real-world cloud security applications. A comparative analysis is shown below with the latest proposed methods:

Study	Technique / Model	Target Environment	Key Contribution	Limitations
Nawaz et al. (2025)[30]	Lightweight Machine Learning Framework	IoT Networks	Designed a lightweight detection model suitable for resource-constrained IoT devices.	Primarily focuses on IoT environments and does not consider detailed traffic activity classification.
Rustam et al. (2025)[31]	Continuous Learning Framework (MULTI-LF)	Multi-environment Networks	Introduced a unified framework that continuously adapts to evolving attack patterns across different environments.	Continuous learning frameworks may increase system complexity and require large computational resources.
de Melo et al. (2025)[32]	Federated GAN-based Detection (Anomaly-Flow)	Distributed Networks	Utilizes federated learning and generative adversarial networks to detect distributed DDoS attacks across domains.	GAN-based systems are computationally expensive and difficult to interpret.

Fathian & Seifousadati (2026)[33]	Real-time Machine Learning Framework	General Network Environment	Developed a real-time DDoS detection and mitigation system capable of detecting attacks during live network operations.	Focuses mainly on real-time deployment but does not address feature optimization or multi-class attack categorization.
Kalafy et al. (2026)[34]	Dynamic Graph Neural Network (DGNN)	Software Defined Networking (SDN)	Uses graph-based deep learning to capture spatial and temporal traffic relationships for improved attack detection.	Graph neural networks require high computational resources and complex traffic representation.
Proposed Method	**AutoML-Optimized Random Forest with Feature Reduction**	**Cloud Computing Environment**	**Performs automated model optimization, feature reduction, and multi-level classification (binary, ternary, and activity-based traffic categories). Achieves high detection accuracy with lower computational complexity.**	**Focused on the cloud network traffic dataset.**

4.4. Our Approach Achieved Better Results

1. Use of a Modern Cloud Dataset-Many earlier studies relied on older or limited datasets that do not represent real cloud traffic behavior. Our work uses a recent cloud-specific dataset that captures dynamic and multi-application traffic patterns, which improves model learning and generalization.
2. Beyond Simple Binary Detection-Most existing research focuses only on distinguishing between benign and attack traffic. Our approach extends to ternary and fine-grained activity classification, allowing the system to detect suspicious behavior and specific attack types more accurately.
3. Automated Model Optimization-Unlike traditional studies that manually tune models, we integrated AutoML with Random Forest. This ensured optimal hyperparameter selection and reduced human bias, leading to consistently higher performance.
4. Effective Feature Reduction-Instead of using all available features, we carefully selected the most relevant 50 features. This reduced noise and redundancy, improved computational efficiency, and enhanced classification accuracy.
5. Fine-Grained Traffic Classification-Unlike most studies that focus only on attack detection, the proposed approach supports binary, ternary, and activity-level classification, enabling deeper analysis of network behavior.
6. Lower Computational Complexity-While deep learning approaches such as graph neural networks or GANs require high computational resources, the Random Forest-based approach achieves comparable or better performance with significantly lower complexity.

5. Conclusion

Detecting DDoS attacks in cloud infrastructures has been a critical topic of research. It has evolved from traditional methods, such as signature-based detection, into more complex models based on machine learning. The method proposed in this paper combines the BCCC-cPacket-Cloud DDoS-2024 dataset with flexible feature selection and AutoML. The top 40 most relevant features were extracted using feature reduction techniques that enhance computational efficiency and detection performance. The proposed framework achieved a 97% accuracy in the classification of binary, ternary, and multi-vector DDoS attacks, showcasing its practical effectiveness in real-world cloud applications. Using AutoML combined with improved feature engineering, the suggested method is scalable and adaptable to a cloud computing environment.

References

[1] Ahmed Shawish, and Maria Salama, *Cloud Computing: Paradigms and Technologies*, Inter-cooperative Collective Intelligence: Techniques and Applications, Springer, pp. 39-67, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[2] Said El Kafhali, and Khaled Salah, "Stochastic Modelling and Analysis of Cloud Computing Data Center," *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)*, Paris, France, pp. 122-126, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Peter Mell, and Tim Grance et al., "*The NIST Definition of Cloud Computing*," National Institute of Standards and Technology, Report, pp. 1-7, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[4] Said El Kafhali, and Khaled Salah, "Performance Analysis of Multi-core VMs Hosting Cloud SaaS Applications," *Computer Standards & Interfaces*, vol. 55, pp. 126-135, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[5] Karthik Lakshminarayanan et al., "Taming IP Packet Flooding Attacks," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 1, pp. 45-50, 2004. [CrossRef] [Google Scholar] [Publisher Link]

[6] Virgil D. Gligor, "A Note on Denial-of-Service in Operating Systems," *IEEE Transactions on Software Engineering*, vol. SE-10, no. 3, pp. 320-324, 1984. [CrossRef] [Google Scholar] [Publisher Link]

[7] C.H. Hoi Steven, Jialei Wang, and Peilin Zhao, "Libol: A Library for Online Learning Algorithms," *The Journal of Machine Learning Research*, vol. 15, no. 15, pp. 495-499, 2014. [Google Scholar] [Publisher Link]

[8] Kenneth R Foster, Robert Koprowski, and Joseph D Skufca, "Machine Learning, Medical Diagnosis, and Biomedical Engineering Research - Commentary," *Biomedical Engineering Online*, vol. 13, pp. 1-9, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[9] Bernard Marr, *A Short History of Machine Learning — Every Manager Should Read*, 2016. [Google Scholar] [Publisher Link]

[10] Ankit Thakkar, and Ritika Lohiya, "A Review of the Advancement in Intrusion Detection Datasets," *Procedia Computer Science*, vol. 167, pp. 636-645, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[11] Tao Peng, Christopher Leckie, and Kotagiri Ramamohanarao, "Survey of Network-based Defense Mechanisms Countering the DoS and DDoS Problems," *ACM Computing Surveys*, vol. 39, no. 1, pp. 1-42, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[12] Jun Xu, and Wooyong Lee, "Sustaining Availability of Web Services under Distributed Denial of Service Attacks," *IEEE Transactions on Computers*, vol. 52, no. 2, pp. 195-208, 2003. [CrossRef] [Google Scholar] [Publisher Link]

[13] Yuri Gil Dantas, Vivek Nigam, and Iguatemi E. Fonseca, "A Selective Defense for Application Layer DDoS Attacks," *2014 IEEE Joint Intelligence and Security Informatics Conference*, The Hague, Netherlands, pp. 75-82, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[14] Mouhammd Alkasassbeh et al., "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, pp. 436-445, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[15] Carl Livadas et al., "Usilng Machine Learning Technliques to Identify Botnet Traffic," *Proceedings 2006 31st IEEE Conference on Local Computer Networks*, Tampa, FL, USA, pp. 967-974, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[16] Bayu Adhi Tama, and Kyung-Hyune Rhee, "Data Mining Techniques in DoS/DDoS Attack Detection: A Literature Review," *International Information Institute*, vol. 18, no. 8, pp. 3739-3747, 2015. [Google Scholar]

[17] Qin Liao et al., "Application Layer DDoS Attack Detection using Cluster with Label based on Sparse Vector Decomposition and Rhythm Matching," *Security and Communication Networks*, vol. 8, pp. 3111-3120, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[18] Peng Xiao et al., "Detecting DDoS Attacks Against Data Center with Correlation Analysis," *Computer Communications*, vol. 67, pp. 66-74, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[19] Reyhaneh Karimazad, and Ahmad Faraahi, "An Anomaly-Based Method for DDoS Attacks Detection using RBF Neural Networks," *2011 International Conference on Network and Electronics Engineering*, Singapore, pp. 44-48, 2011. [Google Scholar] [Publisher Link]

[20] Rui Zhong, and Guangxue Yue, "DDoS Detection System Based on Data Mining," *Proceedings of the 2nd International Symposium on Networking and Network Security*, 2010. [Google Scholar]

[21] Yi-Chi Wu et al., "DDoS Detection and Traceback with Decision Tree and Grey Relational Analysis," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 7, no. 2, pp. 121-136, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[22] Jin Li, Yong Liu, and Lin Gu, "DDoS Attack Detection Based on Neural Network," *2010 2nd International Symposium on Aware Computing*, Tainan, Taiwan, pp. 196-199, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[23] V. Akilandeswari, and S. Mercy Shalinie, "Probabilistic Neural Network based Attack Traffic Classification," *2012 Fourth International Conference on Advanced Computing (ICoAC)*, Chennai, India, pp. 1-8, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[24] Jie-Hao Chen et al., "DDoS Defense System with Turing Test and Neural Network," *2012 IEEE International Conference on Granular Computing*, Hangzhou, China, pp. 38-43, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[25] Hui Li, and Dihua Liu, "Research on Intelligent Intrusion Prevention System based on Snort," *2010 International Conference on Computer, Mechatronics, Control and Electronic Engineering*, Changchun, China, pp. 251-253, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[26] Laheeb Mohammad Ibrahim, "Anomaly Network Intrusion Detection System Based on Distributed Time-Delay Neural Network (DTDNN)," *Journal of Engineering Science and Technology*, vol. 5, no. 4, pp. 457-471, 2010. [Google Scholar] [Publisher Link]

[27] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy*, vol. 1, pp. 108-116, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[28] Cybersecurity Datasets (Intelligence-led Security), Behaviour-Centric Cybersecurity Center (BCCC), Yorku. [Online]. Available: https://www.yorku.ca/research/bccc/ucs-technical/cybersecurity-datasets-cds/

[29] MohammadMoein Shaf et al., "Toward Generating a New Cloud-Based Distributed Denial of Service (DDoS) Dataset and Cloud Intrusion Traffic Characterization, *Information*, vol. 15, no. 4, pp. 1-127, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[30] Mamoona Nawaz et al., "Lightweight Machine Learning Framework for Efficient DDoS Attack Detection in IoT Networks," *Scientific Reports*, vol. 15, pp. 1-24, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[31] Furqan Rustam, Islam Obaidat, and Anca Delia Jurcut, "MULTI-LF: A Continuous Learning Framework for Real-Time Malicious Traffic Detection in Multi-Environment Networks," *arXiv preprint*, pp. 1-23, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[32] Leonardo Henrique de Melo et al., "Anomaly-Flow: A Multi-Domain Federated Generative Adversarial Network for Distributed Denial-of-Service Detection," *IEEE Network*, vol. 40, no. 2, pp. 269-277, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[33] Mohammad Fathian, and Alireza Seifousadati, "A Real-time Machine-learning Model for Detecting and Mitigating DDoS Attacks," *Cybersecurity*, vol. 9, pp. 1-17, 2026. [CrossRef] [Google Scholar] [Publisher Link]

[34] Saad Ahmed Ali Kalafy, Saied Pashazadeh, and Pedram Salehpourge, "Dynamic Graph Neural Network-based Framework to Increase Detection Accuracy in SDN under DDOS," *Scientific Reports*, vol. 16, pp. 1-19, 2026. [CrossRef] [Google Scholar] [Publisher Link]