Journal of Information Systems Engineering and Management
2025, 10(54s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

DDoS Detection by Using Machine Learning

Asmaa A. Alhussain[1], Bassma S. Alsulami[1]
[1]*Computer Science Department, Faculty of Computing and Information Technology,*
King Abdulaziz University, Jeddah, Saudi Arabia
Balsulami@kau.edu.sa

ARTICLE INFO

Received: 30 Dec 2024

Revised: 12 Feb 2025

Accepted: 26 Feb 2025

ABSTRACT

Distributed Denial of Service attack (DDoS) is the most risky attack in network security. DDoS attacks prevent essential services from operating normally for many online applications. With an increasing number of these attacks, the task of detection and mitigation has become increasingly challenging. Among the numerous methods available for detecting Distributed Denial of Service (DDoS) attacks, machine learning techniques have shown great promise in effectively identifying and preventing such attacks. In this project, the machine learning-based model was proposed to detect DDoS attacks. The proposed model used the DDoS-CICIDS2017 dataset with 79 features, and applied four algorithms: Logistic Regression (LR), Support Vector Machine (SVM) with different kernels, Random Forest (RF), and Gradient Boosting (GB). The results highlight the outstanding performance of the Random Forest model, achieving an exceptional 99.99% accuracy, precision, recall, and F1 Score. Notably, this model demonstrated a perfect precision of 100.00%, underscoring its efficacy in accurately classifying DDoS traffic and solidifying its role as a formidable defense against these cyber threats.

Keywords: Distributed Denial of Service Attack, Machine Learning, Random Forest, Support Vector Machine, Gradient Boosting, Logistic Regression

1. INTRODUCTION

In the interconnected world of the internet, the rise of cyber threats is an ever-present challenge. Among the various tactics employed by malicious actors, Distributed Denial of Service (DDoS) attacks stand out as particularly potent and disruptive.

A Denial of Service (DoS) attack can occur through various methods, targeting different systems such as websites, cloud infrastructure, or specific layers of the OSI network model. The two primary forms of DoS attacks are flooding services and crashing services. Flooding involves overwhelming a server with excessive traffic, causing the target system to become extremely slow or unresponsive. In contrast, crashing services focus on exploiting system vulnerabilities to cause it to stop functioning altogether.

A Distributed Denial of Service (DDoS) attack is an advanced and more destructive form of DoS attack. It is harder to detect and defend against because the attack is launched from multiple sources, making it difficult to distinguish between legitimate and malicious IP addresses generating the requests [1].

Unexpectedly, high-profile entities such as government agencies, financial institutions, defense organizations, and military departments have fallen victim to these attacks. Even globally recognized platforms like Facebook, Twitter, WikiLeaks, PayPal, and eBay have experienced service disruptions, leading to financial losses, service degradation, and prolonged unavailability [2, 3].

One primary method for identifying such attacks is through an Intrusion Detection System (IDS). IDSs are designed to detect anomalies or intrusions within a network, using two core techniques: signature-based detection and anomaly-based detection. Signature-based IDS compares real-time traffic against a database of known threats, flagging any matching patterns. On the other hand, anomaly-based IDS builds a model of normal network behavior and identifies any deviations as potential threats. This enables the detection of previously unknown or novel attacks [4]. Despite their usefulness, IDS systems face persistent challenges, particularly with misclassification and low detection accuracy, especially in identifying modern, sophisticated attacks [5]. These limitations can cause security analysts to miss serious threats, highlighting the need for more adaptive and intelligent solutions.

To address these challenges, machine learning (ML) techniques have emerged as effective tools for detecting DDoS attacks. ML models analyze network traffic by extracting meaningful features and training on labeled datasets to recognize patterns indicative of malicious behavior. By incorporating ML algorithms into IDS, systems can better adapt to evolving threats and improve detection accuracy.

In this project, we propose a machine learning-based model for DDoS detection aimed at improving IDS accuracy. The model will be trained and evaluated using various datasets and performance metrics to enhance the identification and classification of DDoS attack patterns through an anomaly-based approach.

This research contributes to the field of network security by developing an intelligent intrusion detection system (IDS) for effectively identifying DDoS attacks. This system leverages machine learning techniques to collect and analyze network traffic data, extracting relevant features and creating labeled training data. The system will evaluate four algorithms to select the model with the best accuracy. The expected research contributions include the development of this IDS, which can enhance network security by effectively detecting and mitigating DDoS attacks, thereby safeguarding critical network services from potential disruption.

2. BACKGROUND

2.1 Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks

A DoS attack aims to render network resources unavailable to legitimate users by overwhelming the target system with malicious traffic. Typically, a DoS attack originates from a single infected device or virtual machine using a standard internet connection. In contrast, a DDoS attack amplifies this disruption by coordinating multiple compromised devices—often part of a botnet—to flood the target from numerous origins, making it significantly more destructive and difficult to mitigate. The primary difference between DoS and DDoS attacks lies in the number of sources. A typical DDoS attack architecture involves four main components: the attacker, control masters (handlers), agents (botnets or zombies), and the victim (target). The attacker scans the internet for vulnerable systems and compromises them to serve as handlers. These handlers, in turn, recruit additional machines (zombies) by installing malicious code. When activated by the attacker, the handler-controlled botnet floods the target, causing service outages. To obscure their identity, attackers often use IP spoofing, allowing the reuse of compromised machines in future attacks [6].

DDoS attacks are commonly categorized based on their nature, target layer, and volume. The three main categories are Protocol Attacks, Application Layer Attacks, and Volumetric Attacks.

1) Protocol or Network-Layer Attacks

Also known as State Exhaustion Attacks, these target network infrastructure components such as firewalls and load balancers. One example is the Smurf Attack, which exploits Internet Control Message Protocol (ICMP) echo requests. The attacker sends spoofed ICMP packets to the broadcast address of a network, with the victim’s IP as the source. Each device on the network replies to the victim, overwhelming it with responses [2].

2) Application Layer Attacks

These attacks overload applications or services by flooding them with legitimate-looking requests. A common example is the HTTP Flood, where an attacker sends HTTP GET or POST requests to exhaust server resources. In incomplete HTTP floods, the attacker only sends partial HTTP headers at regular intervals to keep the connection open, consuming memory and thread resources on the server.

Another example is SQL Injection Distributed Denial of Service (SIDDOS), where attackers inject malicious SQL code through client-side requests (e.g., from browsers), causing excessive load or crashes on the server [2].

3) Volumetric Attacks

These remain among the most prevalent forms of DDoS. The attacker uses massive amounts of data to saturate the target's bandwidth. This results in severe congestion, disrupting normal traffic [3].

2.2 Intrusion Detection Systems (IDS)

Intrusion Detection Systems (IDS) are critical components in modern cybersecurity infrastructures, designed to detect unauthorized access, anomalous behavior, or policy violations within a network or host environment. IDS enhances the ability to identify threats such as Distributed Denial of Service (DDoS) attacks by monitoring and analyzing system activity without requiring continuous human intervention. IDSs are broadly classified based on two dimensions: the data source and the detection methodology. From a source-based perspective, IDSs are categorized as either Host-Based (HIDS) or Network-Based (NIDS). HIDS operates on individual machines, analyzing local system logs and application behavior, whereas NIDS monitors traffic across network segments, detecting broader threats and protocol anomalies. In terms of detection techniques, IDSs are generally classified into Signature-Based (SIDS) and Anomaly-Based (AIDS) systems. SIDS relies on predefined patterns of known threats and offers high precision in detecting established attacks. However, its effectiveness diminishes in the face of zero-day and polymorphic threats. AIDS, in contrast, models normal system behavior and flags deviations as potential threats, offering the capability to detect unknown or emerging attacks. Nonetheless, anomaly detection systems are often challenged by high false-positive rates. To address the limitations of each approach, hybrid intrusion detection systems have been proposed. These systems integrate both signature- and anomaly-based methods to improve accuracy, adaptability, and the overall robustness of threat detection mechanisms.

3. RELATED WORK

The threat of DDoS attacks persists and continues to escalate, posing a significant challenge to network infrastructure and cybersecurity. In this section, we will delve into related research focusing on the critical domain of DDoS attack detection, emphasising the role of machine learning techniques. Rimal et al. [7] used Support Vector Machine (SVM) to predict and detect DDoS attacks in a network with an impressive accuracy rate of 99.68. Arpitha et al. [8] compared the classification performance of four supervised machine learning algorithms: Decision Tree, SVM, Logistic Regression (LR), and K-Nearest Neighbours (KNN). The Jaccard score is used as a metric to assess the accuracy of these algorithms, with Decision Tree achieving a score of 94.3%, SVM scoring 94.4%, Logistic Regression at 94.2%, and KNN at 94.1%. SVM was identified as the most effective algorithm among those evaluated in the study. Saini et al. [9] utilized the WEKA machine learning tool to classify these attacks, comparing the performance of different classifiers, including J48, MLP, Random Forest, and Naïve Bayes. The results indicate that the J48 classifier outperformed the others, achieving an accuracy rate of 98.64%, while MLP, Random Forest, and Naïve Bayes classifiers achieved 98.63%, 98.10%, and 96.93%, respectively. They emphasized the importance of addressing the evolving landscape of DDoS attacks by creating datasets that include a wider range of modern attack types. Azmi et al. [10] addressed two issues: the rise in DDoS attacks and the underutilization of Decision Table classifiers. It overcome these obstacles byidentifying the most relevant features from the UNSW-NB 15 dataset using information gain and data reduction methods. Abbas et al. [11] proposed a methodology for DDos detection by merging datasets from two sources (PORTMAP and LDAP) within the CICDDOS2019 datasets, comprising both attack and benign traffic. They achived the highest accuracy of 99.97%, a 100% detection rate, a minimal false alarm rate, and a robust F-measure of 99.9% when log2 and PCA were employed in the preprocessing stage, demonstrating the effectiveness of this approach in DDOS network attack detection.

4. METHODS

The proposed solution employs machine learning techniques for the detection of potential intrusions. This method will be implemented using the Python programming language and several supervised machine learning algorithms, including Decision Tree (DT), LR, SVM, and GB. The outlined approach consists of the following steps, illustrated in Figure 1.

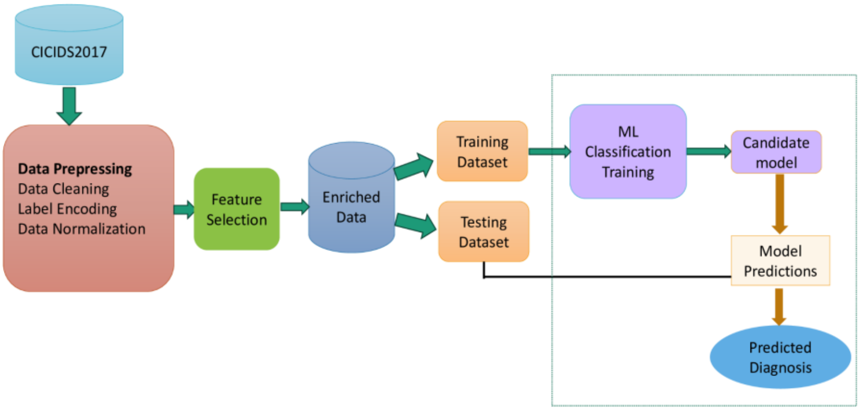
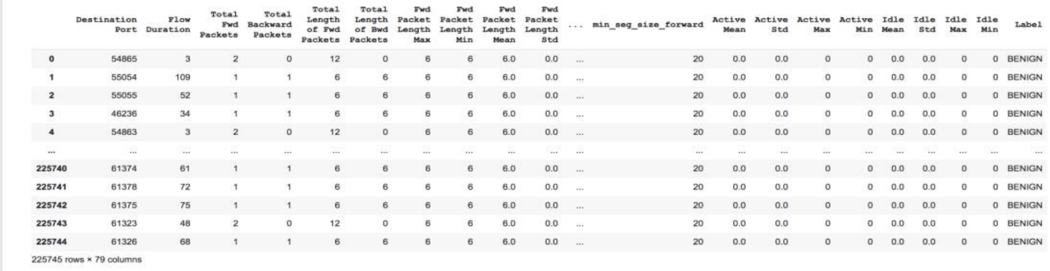


Figure 1: General Approach for the Proposed Method

4.1 Dataset

The CICIDS2017 dataset, generated by the Canadian Institute for Cybersecurity, offers a realistic benchmark for evaluating intrusion detection systems. It captures both normal and malicious network traffic, simulating real-world scenarios through the activities of 25 users interacting over common protocols like HTTP, FTP, SSH, and email services. The dataset includes detailed labels and metadata such as timestamps, IP addresses, ports, protocols, and attack types, stored in PCAP and CSV formats. Generated through human-driven interactions in a testbed, it provides a comprehensive foundation for cybersecurity research. Figure 2 shows the dataset in CSV format.

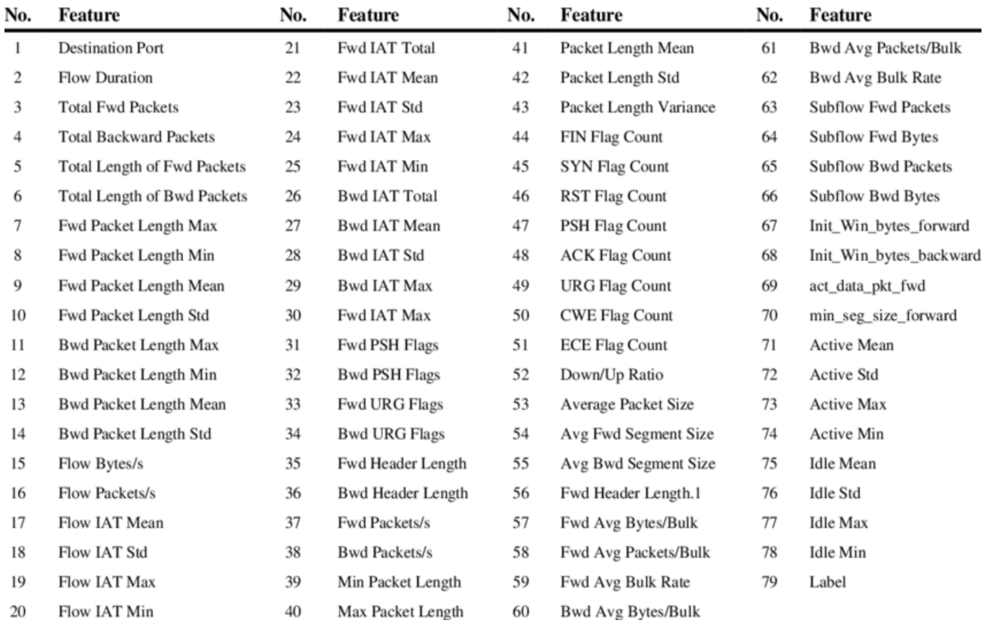


	Destination Port	Flow Duration	Total Fwd Packets	Total Backward Packets	Total Length of Fwd Packets	Total Length of Bwd Packets	Fwd Packet Length Max	Fwd Packet Length Min	Fwd Packet Length Mean	Fwd Packet Length Std	...	min_seg_size_forward	Active Mean	Active Std	Active Max	Active Min	Idle Mean	Idle Std	Idle Max	Idle Min	Label
0	54865	3	2	0	12	0	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
1	55054	109	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
2	55055	52	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
3	46236	34	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
4	54863	3	2	0	12	0	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
225740	61374	61	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
225741	61378	72	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
225742	61375	75	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
225743	61323	48	2	0	12	0	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN
225744	61326	68	1	1	6	6	6	6	6.0	0.0	...	20	0.0	0.0	0	0	0.0	0.0	0	0	BENIGN

225745 rows × 79 columns

Figure 2: CICIDS2017 dataset

Data were collected over five days, beginning on Monday, July 3, 2017, and ending on Friday, July 7, 2017. This dataset consists of 225,745 records with 79 columns (features). Figure 3 shows the list of features.



No.	Feature	No.	Feature	No.	Feature	No.	Feature
1	Destination Port	21	Fwd IAT Total	41	Packet Length Mean	61	Bwd Avg Packets/Bulk
2	Flow Duration	22	Fwd IAT Mean	42	Packet Length Std	62	Bwd Avg Bulk Rate
3	Total Fwd Packets	23	Fwd IAT Std	43	Packet Length Variance	63	Subflow Fwd Packets
4	Total Backward Packets	24	Fwd IAT Max	44	FIN Flag Count	64	Subflow Fwd Bytes
5	Total Length of Fwd Packets	25	Fwd IAT Min	45	SYN Flag Count	65	Subflow Bwd Packets
6	Total Length of Bwd Packets	26	Bwd IAT Total	46	RST Flag Count	66	Subflow Bwd Bytes
7	Fwd Packet Length Max	27	Bwd IAT Mean	47	PSH Flag Count	67	Init_Win_bytes_forward
8	Fwd Packet Length Min	28	Bwd IAT Std	48	ACK Flag Count	68	Init_Win_bytes_backward
9	Fwd Packet Length Mean	29	Bwd IAT Max	49	URG Flag Count	69	act_data_pkt_fwd
10	Fwd Packet Length Std	30	Fwd IAT Max	50	CWE Flag Count	70	min_seg_size_forward
11	Bwd Packet Length Max	31	Fwd PSH Flags	51	ECE Flag Count	71	Active Mean
12	Bwd Packet Length Min	32	Bwd PSH Flags	52	Down/Up Ratio	72	Active Std
13	Bwd Packet Length Mean	33	Fwd URG Flags	53	Average Packet Size	73	Active Max
14	Bwd Packet Length Std	34	Bwd URG Flags	54	Avg Fwd Segment Size	74	Active Min
15	Flow Bytes/s	35	Fwd Header Length	55	Avg Bwd Segment Size	75	Idle Mean
16	Flow Packets/s	36	Bwd Header Length	56	Fwd Header Length.1	76	Idle Std
17	Flow IAT Mean	37	Fwd Packets/s	57	Fwd Avg Bytes/Bulk	77	Idle Max
18	Flow IAT Std	38	Bwd Packets/s	58	Fwd Avg Packets/Bulk	78	Idle Min
19	Flow IAT Max	39	Min Packet Length	59	Fwd Avg Bulk Rate	79	Label
20	Flow IAT Min	40	Max Packet Length	60	Bwd Avg Bytes/Bulk		

Figure 3: Features of the CICIDS2017 dataset

The distribution of the classes is as follows: benign data with 97,718 and DDoS attack data with 128,027 observations. In the end, the dataset was split into 70% for training and 30% for testing for model evaluation.

4.2 Data Preprocessing

This research focuses on a binary classification, where each observation is designated as normal or indicative of an attack. Before initiating the training of the IDS model, the chosen data sets underwent subsequent preprocessing steps. Data cleaning is the initial phase of data preprocessing to remove noise, fill in missing values, and rectify discrepancies. To ensure data consistency, integrity, readability, and compatibility with analysis tools, we removed any leading or trailing white spaces within the dataset. To avoid bias, we eliminated all duplicate records from the data, retaining only one copy of each record. After this operation, 2633 rows were removed from the dataset. Samples containing NaN and INF values in the 'Flow Bytes/s' and 'Flow Backest/s' features were removed from the dataset. Columns with constant values do not contribute to learning but increase the data dimension. Consequently, features with constant values were removed from the CIC-IDS2017 (DoS) dataset, resulting in the removal of ten features (see Table 1).

Table 1: List of features with constant values

Bwd PSH Flags	Fwd URG Flags
Bwd URG Flags	CWE Flag Count
Fwd Avg Packets/Bulk	Fwd Avg Bytes/Bulk
Fwd Avg Bulk Rate	Bwd Avg Bytes/Bulk
Bwd Avg Packets/Bulk	Bwd Avg Bulk Rate

After data cleaning steps, we implemented label encoding to convert non-numerical categories to numeric categories. We assigned instances corresponding to anomalous traffic as '1' and instances representing normal traffic as '0'. Subsequently, we implemented Standardization, or Z-score scaling, which is a preprocessing method used in machine learning to rescale data so that each feature has a mean of zero and a standard deviation of one. This technique helps improve the performance of algorithms that are sensitive to the scale of input features, such as logistic regression, SVMs, and neural networks.

4.3 Feature Selection

Feature selection helps improve model efficiency by keeping only the most useful data while removing features that add little or no value. In this process, the Correlation-based Feature Selection (CFS) method was used to find features that are strongly linked to the target outcome but not closely related to each other. This reduces noise in the data and helps the model focus on what truly matters. The heatmap presented in Figure 4 illustrates the relationships among various numerical features, providing valuable insights that inform the decision-making process regarding which features should be retained or eliminated.

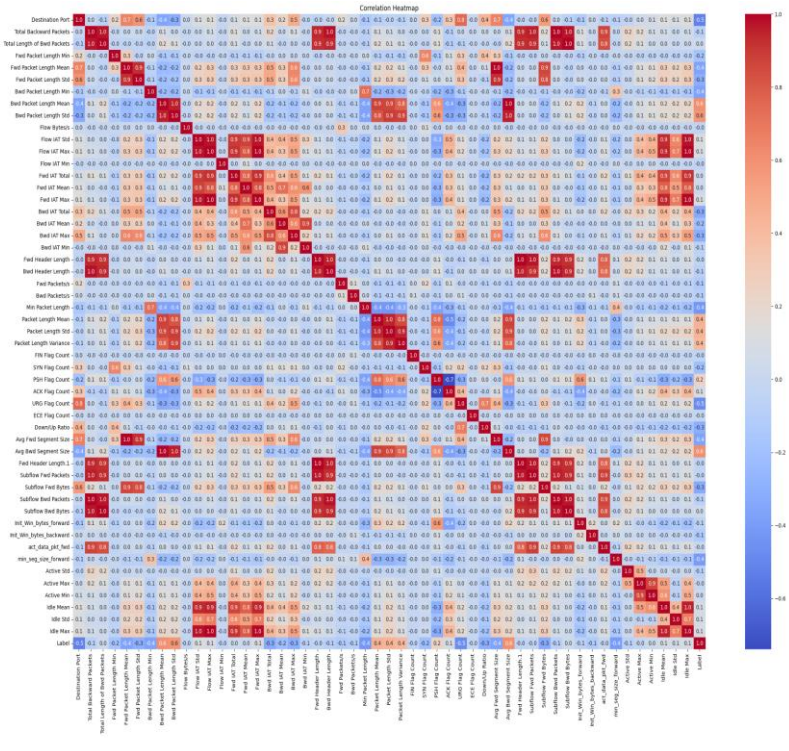


Figure 4: Heatmap Correlation between all numerical variables

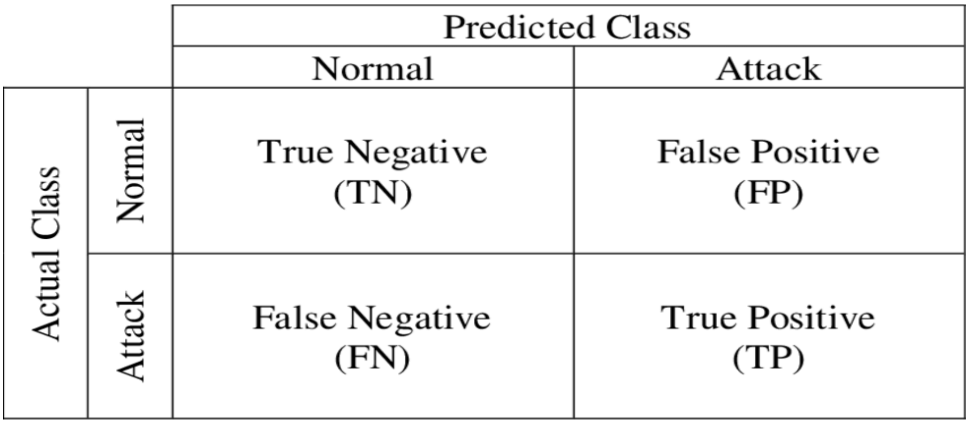
4.4 Machine Learning Classification Model

Machine learning is a method where computers use data to learn patterns and make decisions with little manual guidance. Its rise is closely linked to the expansion of large datasets, cheaper storage, and faster computing. One common approach, supervised learning, works with data that includes both inputs and expected results, allowing the system to learn how the two are connected. A simple example would be teaching a model to estimate beach crowd sizes based on daily temperature records. The training process consists of selecting a machine learning algorithm, fitting the model to the training data, and adjusting the model's parameters to minimize the difference between its predictions and the actual target values in the training set. In this study, we implemented four ML algorithms, including SVM, RF, GB, and LR.

SVM is a robust classification approach that determines the optimal hyperplane for separating classes and is particularly effective in both linear and non-linear contexts due to its use of various kernel functions. Its strength lies in handling high-dimensional data and modeling complex decision boundaries with precision. Random Forest, an ensemble technique, improves model resilience and accuracy by aggregating predictions from multiple decision trees. It is widely used in both classification and regression tasks, offering strong resistance to overfitting and the ability to manage large, complex datasets. Gradient Boosting, another ensemble approach, constructs models in a sequential manner, each iteration correcting the errors of its predecessor. Though computationally demanding, it is highly effective for tasks where predictive accuracy is critical. Logistic regression remains a popular choice for binary and multi-class classification problems, valued for its interpretability and efficiency, particularly when the relationship between variables is linear.

5. RESULTS

The evaluation of ML algorithms involves calculating their accuracy in making predictions on data they have not previously encountered. After training, the model's output—whether class labels or probability scores—is tested using appropriate performance metrics. Several criteria are employed to assess the accuracy and reliability of classification tasks. For binary classification, key metrics include Confusion Matrix, Accuracy, Precision, Recall, and F1-Score, each emphasising distinct aspects of the model's predictive ability. The confusion matrix serves as a common method for evaluating binary classification. It illustrates the predicted values for both classes against the actual values of the classes. In Figure 5, True Positives (TP), False Positives (FP), False Negatives (FN), and True Negatives (TN) are presented in the confusion matrix. In binary classification, where there are two classes—one considered positive and the other negative—TP and TN represent the items correctly classified for each class. On the other hand, FP and FN correspond to the elements incorrectly classified. Unlike accuracy, the confusion matrix provides a visual representation of how the model predicts both classes.



		Predicted Class	
		Normal	Attack
Actual Class	Normal	True Negative (TN)	False Positive (FP)
	Attack	False Negative (FN)	True Positive (TP)

Figure 5: Confusion Matrix

A frequently used metric is accuracy (Equation 1), computed by taking the ratio of the sum of true positive and true negative values (instances correctly classified) to the total number of instances in the dataset, encompassing both positive and negative cases. Precision (Equation 2) represents the proportion of samples accurately classified for a specific class relative to the total number of elements classified for that class. The recall metric (Equation 3), also known as sensitivity, quantifies the number of items correctly classified for a specific class relative to the total number of samples for that class. The F1-score (Equation 4) is a composite metric that incorporates both precision and recall.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad (1)$$

$$Precision = \frac{TP}{TP + FP} \qquad (2)$$

$$Recall = \frac{TP}{TP + FN} \qquad (3)$$

$$F1 - Score = 2\,X\,\frac{Precision\ X\ Recall}{Precision + Recall} \qquad (4)$$

Table 2 shows the performance metrics, accuracy (Acc), precision (Pr), recall (Re), and F1-score (F1) for LR, four kernels of SVM, RF, and GB. The LR algorithm demonstrates robust performance with an accuracy of 98.98%, highlighting its effectiveness in accurately classifying instances. The precision of 98.35% underscores the model's accuracy in predicting DDoS instances, while a recall of 99.96% showcases its capability to capture almost all actual DDoS instances. The F1 Score, representing the harmonic mean of precision and recall, is 99.15%, indicating a well-balanced performance. The utilization of the 'liblinear' solver contributes to the optimization of the logistic regression process. After evaluating the performance metrics for different SVM kernels, the linear kernel stands out as the best choice. The algorithm demonstrates exceptional accuracy at 99.91%, high precision of 99.93%, recall of 99.92%, and F1-score of 99.93% for both classes. The Random Forest model showcases outstanding performance with an accuracy, precision, recall, and F1 Score all reaching an impressive 99.99%. The precision of 100.00% for both classes indicates the model's precision in correctly identifying instances, supported by a near-perfect recall. The Gradient Boosting model demonstrates exceptional performance with an accuracy, precision, recall, and F1-Score all reaching 99.99%. It achieves a balanced and high level of performance, making it the most effective model for the given classification task.

Table 2: Performance Metrics for LR, SVM, RF and GB

Model	Accuracy (%)	Precision (%)	Recall (%)	
LR	98.98	98.35	99.96	99.15
SVM (Polynomial)	99.01	98.84	99.15	98.98
SVM (RBF)	99.89	99.98	99.98	99.98
SVM (Sigmoid)	96.67	96.56	96.63	96.56
SVM (Linear)	99.91	99.93	99.92	99.93
RF	99.99	100	99.99	99.99
GB	99.99	99.99	99.99	99.99

The confusion matrix shown in Figure 6 illustrates that the Random Forest model has an exceptionally low number of misclassifications, with a high number of true positives and true negatives. This reinforces the model's robustness in correctly classifying instances, both positive and negative, contributing to its outstanding accuracy and precision.

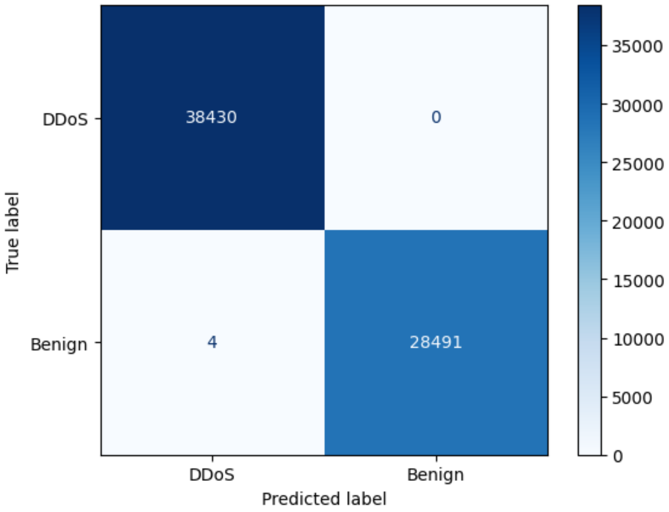


Figure 6: Confusion Matrix for RF

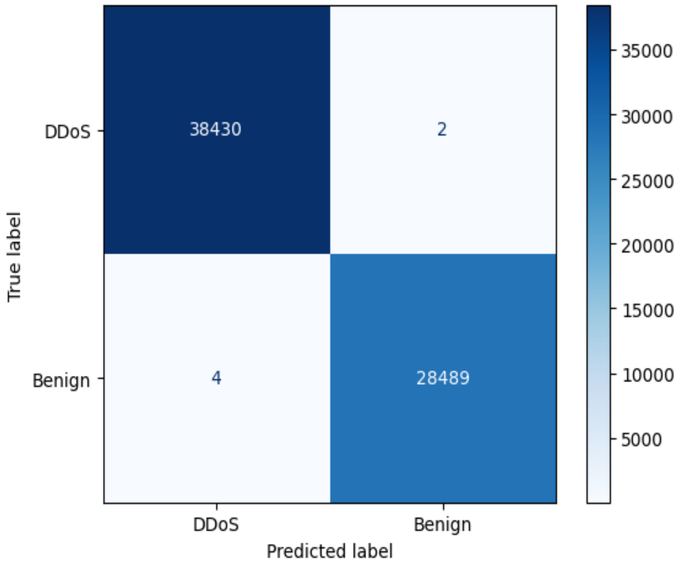


Figure 7: Confusion Matrix for GB

The confusion matrix shown in Figure 7 reinforces the model's reliability, with only 2 false DDos and 4 false Benign among the 66,925 instances. This outstanding accuracy suggests the model's proficiency in correctly classifying instances, contributing to a robust predictive capability. The minimal discrepancies in precision and recall indicate a well-balanced model.

6. CONCLUSION AND FUTURE WORK

This study explores the application of ML techniques for DDoS attack detection using a binary classification approach. The presented model was trained and evaluated on the CICIDS2017-DDoS dataset to distinguish between legitimate and malicious network traffic. Several classification algorithms were tested, including RF, GB, SVM, and LR. Among these, Random Forest demonstrated the highest level of performance, achieving near-perfect scores across all evaluation metrics. While the other models also showed strong results, they were marginally outperformed by the ensemble-based methods. The findings highlight the potential of machine learning in enhancing intrusion detection systems, and future work will focus on expanding the model's capabilities to classify different types of DDoS attacks, incorporating diverse algorithms, and validating performance across varied datasets to improve robustness and adaptability.

REFRENCES

[1] Vishal Verma and Vasudha Kumar. Dos/ddos attack detection using machine learning: A review. In Proceedings of the International Conference on Innovative Computing & Communication (ICICC), 2021.

[2] Irfan Sofi, Amit Mahajan, and Vibhakar Mansotra. Machine learning techniques used for the detection and analysis of modern types of ddos attacks. Int. Res. J. Eng. Technol, 4(6):1085–1092, 2017.

[3] M Arshi, MD Nasreen, and Karanam Madhavi. A survey of ddos attacks using machine learning techniques. In E3S Web of Conferences, volume 184, page 01052. EDP Sciences, 2020.

[4] Ramin Fadaei Fouladi, Cemil Eren Kayatas, and Emin Anarim. Frequency based ddos attack detection approach using naive bayes classification. In 2016 39th International Conference on Telecommunications and Signal Processing (TSP), pages 104–107. IEEE, 2016.

[5] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. applied sciences, 9(20):4396, 2019.

[6] A Srivastava, BB Gupta, A Tyagi, Anupama Sharma, and Anupama Mishra. A recent survey on ddos attacks and defense mechanisms. In International Conference on Parallel Distributed Computing Technologies and Applications, pages 570–580. Springer, 2011.

[7] A. N. Rimal and R. Praveen. Ddos attack detection using machine learning. Journal of Emerging Technologies and Innovative Research, 2020.

[8] KS Arpitha, Mrs K Hema, G Sona, S Koushik Reddy, and Punith Babu GU. Ddos attacks using machine learning. J. Xi'an Univ. Archit. Technol, 2(4):3380–3384, 2020.

[9] Parvinder Singh Saini, Sunny Behal, and Sajal Bhatia. Detection of ddos attacks using machine learning algorithms. In 2020 7th International Conference on Computing for Sustainable Global Development (INDIACom), pages 16–21. IEEE, 2020.

[10] M. A. H. Azmi, C. F. M. Foozy, K. A. M. Sukri, N. A. Abdullah, I. R. A. Hamid, and H. Amnur. Feature selection approach to detect ddos attack using machine learning algorithms. Journal Name, Volume Number:Page Numbers, 2021.

[11] S. A. Abbas and M. S. Almhanna. Distributed denial of service attacks detection system by machine learning based on dimensionality reduction. Journal of Physics: Conference Series, 1804:012136, 2021.