International Conference on Science, Engineering & Management Trends

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

DDoS Attack Detection Using Network Traffic Features and Machine Learning

Vaishnavi Singh, Harsh Kumar Singh , Shreya Singh , Rajat Takkar
Department of Computer Science and Engineering Chitkara University Institute of Engineering and Technology, Chitkara University Rajpura, Punjab, India

Abstract- DDoS attacks pose a significant risk to contemporary network infrastructure. By overloading network resources with malicious traffic, they cause service disruptions. Conventional intrusion detection systems frequently fall short in the face of dynamic and frequent DDoS attacks because they depend on established patterns. In order to identify DDoS attacks, this paper presents a machine learning technique that examines network traffic characteristics. Due to data leakage, we removed the Source IP attribute from a dataset consisting of 852,585 instances. We used stratified train-test splitting in conjunction with label encoding to encode categorical characteristics. Packet Length (≈ -0.92) and Destination Port (≈ -0.45) were identified by correlation analysis as the critical characteristics for identifying attack traffic. XGBoost, Random Forest, and Logistic Regression were the three classifiers that we evaluated. The accuracy of Random Forest and XGBoost was 0.999947 and 0.999953, respectively, while Logistic Regression achieved 0.993432. The findings demonstrate that when combined with appropriate preprocessing and feature analysis, ensemble models provide incredibly precise and dependable DDoS detection.

Keywords— DDoS detection, machine learning, network traffic analysis, Random Forest, XGBoost, cybersecurity.

I. INTRODUCTION

Modern networks are seriously threatened by Distributed Denial of Service (DDoS) assaults. They cause service disruptions by flooding network resources with malicious traffic. Due to their inability to identify unfamiliar patterns, traditional signature-based intrusion detection systems frequently fall short against dynamic and frequent DDoS attacks. A viable alternative is offered by machine learning (ML) algorithms, which accurately distinguish benign and malicious flows by learning traffic behaviour [1]. The attack surface for hackers has greatly expanded due to the quick growth of internet-connected products and services. These days, DDoS attacks are more advanced and frequently target government infrastructure, healthcare systems, and financial organisations. These assaults cause significant monetary losses as well as reputational harm. DDoS attacks are becoming more frequent and larger every year, according to recent cybersecurity reports. In order to preserve network availability and integrity, robust detection systems are essential [2]. These security issues have been made worse by the Internet of Things (IoT). Smart cities, which gather and evaluate data for public utility services via IoT sensors, are particularly vulnerable. Smart utility meters, smart transportation systems, smart air quality monitors, CCTV cameras, and smart medical equipment are examples of devices that facilitate tasks but also provide several ports

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

of entry for hackers [3]. A network of sensors that use IoT and communicate data is called a "smart city," but this interconnection presents significant security challenges [4]. In network security, DDoS is the most prevalent kind of cyberattack. By preventing legitimate users from accessing servers, attackers can cause serious harm [5]. In contemporary network environments, there are two main security issues. The first is how to determine the sort of attack in a cloud center, particularly given the numerous covert attacks present in the system. The second is identifying the most effective machine learning techniques to identify attack risks prior to system breach [6]. Because machine learning can automatically identify patterns in data and adjust to new assault types, it has shown itself to be a useful tool for intrusion detection. ML-based techniques can identify anomalous behaviour based on statistical differences from typical traffic profiles, in contrast to signature-based systems that depend on established attack patterns [7]. In DDoS detection, where attack patterns are constantly evolving to evade conventional defences, this is very helpful. Several machine learning techniques, such as Random Forest (RF), Decision Trees (DT), Naive Bayes (NB), and deep learning, have been studied recently for DDoS detection. Because they aggregate several weak learners into a single strong learner, these experiments demonstrate that ensemble techniques typically perform better than single classifiers [8]. However, feature selection, data quality, and preprocessing methods have a significant impact on model performance. An essential part of any ML process is data preprocessing. Model performance can be significantly impacted by issues such as missing values, categorical variables, inconsistent data types, and data leakage if they are not properly addressed [9]. By identifying the most crucial characteristics, correlation analysis and feature selection assist lower dimensionality and increase efficiency and accuracy. This paper describes an ML-based technique that uses network traffic characteristics to identify DDoS attacks. The analysis of a dataset containing 852,585 instances aims to achieve the following: Eliminate data leakage characteristics (Source IP) to guarantee model generalisation.

To preserve class distribution, use stratified train-test splitting and label encoding for categorical attributes. Conduct correlation analysis to find important characteristics. Three classifiers are evaluated: XGBoost, Random Forest, and Logistic Regression: Evaluate model performance using classification metrics, accuracy, and confusion matrix Packet Length (≈ -0.92) and Destination Port (≈ -0.45) are the most important characteristics for differentiating attack traffic, according to correlation studies. The accuracy of Logistic Regression is 0.993432, whereas the accuracy of Random Forest and XGBoost is 0.999947 and 0.999953, respectively. This suggests that when combined with efficient preprocessing and feature analysis, ensemble models may accurately and consistently identify DDoS attacks. The following is how the remainder of the paper is arranged: Section II examines current machine learning techniques for DDoS detection. The suggested approach and dataset are described in Section III, together with the model setups and preprocessing procedures. The experimental data are discussed and a comparative analysis is given in Section IV. Section V ends with recommendations for further research.

II. LITERATURE REVIEW

DDoS attack detection using machine learning (ML) and deep learning (DL) has been the subject of numerous research studies in recent years. In chronological order, current approaches, datasets, and techniques are reviewed in this section along with their advantages and disadvantages.

A.Initial Machine Learning Methods for DDoS Identification

Using the BoT-IoT dataset, Sharma and Babbar [1] developed a DDoS detection system for smart cities. Naive Bayes (NB), Random Forest (RF), and Decision Tree (DT) were evaluated. With an accuracy of 91%,

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

RF and DT outperformed NB, which had a score of 71%. The significance of feature extraction and preprocessing for IoT security was emphasised by this study. Saghezchi et al. [2] used real manufacturing network traffic and machine learning to identify anomalies in Industry 4.0. They tried eleven algorithms and extracted forty-five bidirectional flow features. The benefit of using real-world data over synthetic datasets was demonstrated by supervised models, particularly DT and RF, which achieved an accuracy of 99.9% with a false positive rate of 0.1%. A two-stage DDoS detection technique that comprised feature extraction and classification was developed by Pei et al. [3]. They found unique traffic patterns for TCP, UDP, and ICMP floods by examining attack packets from popular tools like TFN2K. They outperformed SVM by using RF to detect TCP/UDP floods with over 98% detection rates.

B. Managing Errors in Data:

The detection of DDoS attacks using suboptimal data conditions was discussed by Dremov and Volokyta [4], who simulated the challenges in practical environments such as
noise, missing values, and bit flipping through the use of the CIC-DDoS2019 dataset to solve issues of class imbalance. ADASYN sampling and Tomek links were two types of complex models used include XGBoost and CatBoost, which were more robust despite losing some level of performance in corrupted data; DNNs achieved an accuracy of 97%.

C. Deep Learning Developments in SDN Environments:

The ML and DL approaches employed to detect DDoS attacks in Software-Defined Networks (SDN) have been discussed by Ali et al. [5]. These approaches have been grouped as either supervised, semi-supervised, or hybrid learning approaches following an analysis of studies carried out between 2018 and 2022. Accuracy greater than 99% was consistently achieved using CNN, DNN, and CNN-LSTM approaches using CICIDS2017 dataset. The limitations identified included use of large balanced datasets, lack of preprocessing, and inability to deploy in real time. A Deep Neural Network (DNN) with transfer learning has been suggested by Saito et al. [6] as an approach to enhance DDoS attack detection accuracy in scenarios where labelled data are limited. In this work, it was observed that SHAP approach was better than RF and MI for feature selection. It had Matthews Correlation Coefficient of 0.951 and Balanced Accuracy of 0.975 using only 200 labelled data.

D. Analysing Comparative Datasets:

BoT-IoT [7] is a standard dataset which is often employed for DDoS attacks and comprises diverse DDoS, DoS, scanning, and information stealing attacks on UDP, TCP, and HTTP protocols.
-CIC-DDoS2019 [8]: This dataset consists of 125,000 records of various attacks and normal flows using CICFlowMeter software for seven distinct types of attacks.
-CICIDS2017 & CSE-CIC-IDS2018 [9]: It presents realistic scenarios including DDoS, botnet, Heartble

E. Research Gaps Identified:

Some of the areas that require attention include [1] - [9]:

1.Data Inconsistencies: Many studies [1-3,5-6] use very carefully curated datasets that do not address many issues like packet losses, data corruption, and noise. While [4] addresses such concerns, controlled experimental setups do not reflect the unpredictability of the actual environment.

2.Class Skewness: Class imbalances can affect the performance of models since the DDoS attacks usually overwhelm normal traffic during benchmarking [4-5,8]. Further evaluation for resampling methods like ADASYN [4] is needed.

3.Features Selection: While feature selection based on SHAP [6] is better than traditional approaches, it should be tested on different datasets [7-9].

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

4.Real-Time Deployments: Most studies evaluate offline datasets [1-6]; however, aspects like latency, throughput, and concept drift in real-time traffic need more evaluation [5].

5. Transfer Learning and Domain Adaptation: The accuracy of models based on benchmark datasets [7-9] may decrease once they interact with real-life traffic. There is a potential for transfer learning [6], however, more investigation is needed.

6. Diversity and Generalization of Datasets: Currently existing datasets do not account for variations in traffic and attacks.There are no cross-dataset validations.

7. Resource Limitations: Application of DL models with high accuracy [5-6] on edge computing and Internet of Things devices is impossible because of high resource demands.

8. Binary or Multiclass Classification: Most research efforts [1-4] focus on binary classification; this approach does not provide sufficient detail when it comes to detecting various types of DDoS attacks, such as SYN or UDP floods.

9. Adversarial Attacks: Little information is available about adversarial models' vulnerability to escape strategies.

TABLE I

Reference	Year	Methodology	Dataset	Metrics	Performance
Rios et al. [1]	2021	Fuzzy Logic, MLP,KNN, SVM,MNB, ED	Emulat-ed & real traffic	Precision, Recall, F1score	MLP: 98.04% F1 (attack); Hybrid: 100% F1 (real traffic)
Doshi et al. [2]	2018	K-NN, SVM, DT, RF, NN	IoT network traffic	Accuracy, Precision, Recall,F1	All >99% accuracy; RF & NN best
Hamarsh eh et al. [3]	2023	RF, DT, SVM, XGBoost	CICDDoS2019	Accuracy, Precision, Recall,F1	RF: 68.9% accuracy
Kar Suvra [4]	2025	LR, K-NN, RF, SVM, NB, AdaBoost, XGBoost, DT	CICDDoS2019	Accuracy, AUC, Execution Time	RF: 98.80% accuracy; DT: 2.015s fastest
Shohan et al. [5]	2024	Hybrid (1D CNN + RF + MLP)	CICDDoS2019	Accuracy, Precision, Recall,F1	Hybrid: 94% all metrics; CV: 0.94 consistent

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

III. MATERIALS AND METHOD

A. Dataset

In this research, the researchers have used an open-source database of DDoS traffic consisting of 852,585 data and including nine features altogether. Since this data includes benign as well as different forms of DDoS attack traffic, it is apt for implementing the task of binary classification. Each individual sample in the dataset consists of a network flow along with its associated characteristics based on packet header information. Table II presents a complete list of features.

TABLE II. FEATURE DESCRIPTION

Feature Name	Description	Data Type
Highest Layer	Highest protocol layer in the network flow	Categorical
Transport Layer	Transport protocol used (TCP/UDP)	Categorical
Dest IP	Destination IP address	Categorical
Source Port	Source port number	Numerical
Dest Port	Destination port number	Numerical
Packet Length	Length of packets in the flow	Numerical
Packets/Time	Packet rate over time	Numerical
Target	0 for normal traffic, 1 for DDoS attack	Binary

B. Data Preprocessing

The process of data preprocessing is an indispensable part of any machine learning procedure. Data preprocessing converts data into a suitable form to enable the model's training. The data was preprocessed for model training through such steps as follows:

1. Removal of Features Containing Data Leakage: The 'Source IP' feature had a perfect correlation with the target variable, according to a preliminary analysis. Such a situation could be indicative of a data leakage problem. Machine learning models depend too heavily on the feature which has a perfect correlation with the target in lieu of discovering patterns within other network features. Unrealistic performance predictions for new data would be one of the possible outcomes. In order to facilitate generalization of the model, the 'Source IP' feature was removed from the dataset.

2. Conversion of Categorical Features to Numerical Form: All machine learning algorithms work with numeric inputs. Thus, any categorical feature should be transformed into a numeric one. In order to transform categories into numeric representations, label encoding was applied to the category columns. The mapping was saved for future reference. Examples of such category columns are 'Highest Layer', 'Transport Layer', and 'Dest IP'.3. Correlation study: To identify the best characteristics for identifying DDoS attacks, a correlation study was carried out. For every feature, the Pearson correlation coefficient with respect to the target variable was computed. When it comes to differentiating between attack and normal traffic, features with high absolute correlation values are seen to be more important. The formula for calculating the correlation coefficient r between feature x and target y is:

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\left((\sum_{i=1}^{n}(x_i - \bar{x})^2)(\backslash sum_{i=1}^{n}(y_i - \bar{y})^2)\right)}}$$

1. Train-Test Split: To maintain the initial class distribution, the preprocessed dataset was split into training and testing sets using stratified sampling. By using stratification, training and testing sets are guaranteed to have about the same proportion of samples from each class as the original dataset. For imbalanced datasets, this is essential to provide accurate model evaluation. A training set (80%, 682,068 instances) and a testing set (20%, 170,517 instances) were separated from the dataset.
2. Feature Scaling: This was done for algorithms that are sensitive to feature magnitudes, such as neural networks and Logistic Regression. Features were standardised to have a unit variance and zero mean. The equation for standardisation is:

$$x_{scaled} = \frac{x - \mu}{\sigma} \quad (1)$$

where μ represents the feature's mean and σ its standard deviation. To avoid data leakage from the testing set into the training process, the scaler was fitted only on the training data and then applied to both training and testing sets.

C. Machine Learning Techniques:

Random Forest, XGBoost, and Logistic Regression were the three machine learning algorithms selected for assessment. These algorithms provide a comprehensive comparison of several machine learning techniques, including gradient boosting, ensemble methods, and linear models.

1.LogisticRegression:
One supervised learning technique for binary classification issues is logistic regression. It is a classifier that models the likelihood that an instance belongs to a particular class, notwithstanding the name. To get a probability score between 0 and 1, the algorithm applies a logistic function to a linear combination of input features. The sigmoid, or logistic function, is described as:

P(y=1|x) = 1 / (1 + e^-(w^T x + b)) (2)

where b denotes the bias term, x the input feature vector, and w the weight vector. The log-loss function is minimised to train the model:

$$L(w, b) = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log(\hat{y_i}) + (1 - y_i)\log(1 - \hat{y_i})] \quad (3)$$

To make sure the logistic regression for this study converged, a maximum of 200 iterations was chosen. To forecast the likelihood of DDoS assaults, the algorithm was trained using the scaled features.

2. The Random Forest:
An ensemble learning technique called Random Forest builds several decision trees during training and outputs the class that is most prevalent among the various trees. It creates an ensemble of independent decision trees using bagging and random feature selection.
The Random Forest forecast for a given input x is:

$\hat{y}$ = majority vote$\{h_1(x), h_2(x), \ldots, h_n(x)\}$(4)

where $h_i(x)$ is the prediction of the i-th decision tree
. For DDoS detection, Random Forest offers a number of benefits. Compared to single decision trees, it is less prone to overfit, robust against noise and outliers, and adept at handling high-dimensional

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

data. It also offers feature importance rankings. For reproducibility, Random Forest was used in this investigation using 100 estimators (trees) and a random state of 42.

3. XGBoost:

Designed to be effective, adaptable, and portable, XGBoost (Extreme Gradient Boosting) is an optimised gradient boosting library. It provides parallel tree boosting and applies machine learning techniques within the Gradient Boosting framework. An ensemble of decision trees is constructed sequentially by XGBoost, with each new tree correcting the mistakes of the previous trees. A training loss and a regularisation term make up the objective function that XGBoost minimises:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k) \quad (5)$$

where l is a differentiable convex loss function measuring the difference between prediction $\hat{y_i}$ and target y_i, and Ω penalizes the complexity of the model:

$$\Omega(f) = \gamma T + \left(\frac{1}{2}\right)\lambda||w||^2 \quad (6)$$

T is the number of leaves in the tree, w is the weight of each leaf, γ is the minimal loss reduction needed to split a leaf node, and λ is the L2 regularisation parameter.

The important parameters utilised in this study are: n_estimators = 100, max_depth = 6, learning_rate = 0.1, subsample = 0.8, colsample_bytree = 0.8, and random_state = 42 to make sure the results can be repeated.

D. Setting Up the Experiment:

We used Python 3.12 and the following libraries to do the experiments: NumPy is used for math, Pandas is used for working with and analysing data, Scikit-learn is used for Logistic Regression, Random Forest, and evaluation metrics, XGBoost is used for the XGBoost classifier, and Matplotlib and Seaborn are used for charting results and visualising data. The trials were done in a planned way. This included loading data from CSV format with Pandas, preprocessing that involved getting rid of data leakage, categorical encoding, and feature scaling, splitting the dataset with stratification, training models with the best hyperparameters, using different performance metrics to evaluate models, and using visualisations to analyse the results. To make sure that the results could be repeated, all of the studies employed a fixed random seed (42).

E. Metrics for Evaluating Performance:

Standard classification metrics from the confusion matrix were used to judge how well the machine learning models worked. The confusion matrix puts the results of predictions into four groups: True Positive (TP) means that attack traffic was correctly identified as an attack, True Negative (TN) means that normal traffic was correctly identified as normal, False Positive (FP) means that normal traffic was incorrectly identified as an attack, and False Negative (FN) means that attack traffic was incorrectly identified as normal. Using these numbers, the following metrics were figured out:

1. Accuracy: Accuracy is the ratio of accurately predicted occurrences to total instances. It tells you how correct the model is overall.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (7)$$

2.Precision: Precision, which is also known as Positive Predictive Value, tells you how many of the positive identifications were truly correct [9].

$$\text{Precision} = \frac{TP}{TP + FP} \quad (8)$$

3.Recall: Recall (also known as Sensitivity or True Positive Rate) is a measure of how many real positives were accurately detected [9].

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

$$\text{Recall} = \frac{TP}{TP + FN} \quad (9)$$

4.F1-Score: The F1-Score is the harmonic mean of Precision and Recall, which means it gives you one number that balances both issues [10].

$$F1\text{-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (10)$$

5.Confusion Matrix Visualization: Heatmaps were used to depict confusion matrices in a way that made it easy to assess how well the model did across classes, showing both right classifications and mistake patterns.

These metrics give a complete picture of how well the model is working. Accuracy is a general measure. Precision tells you how reliable favourable predictions are. Recall demonstrates how well the model can find attacks. The F1-score strikes a balance between these two issues. These metrics are very important for assignments that have unbalanced classification. It can be misleading to just rely on accuracy.

IV. RESULTS AND DISCUSSION

The investigation and assessment of machine learning models used to identify DDoS attacks are covered in this section. We evaluate the accuracy, precision, recall, and F1-score of the Random Forest, XGBoost, and Logistic Regression classifiers. In order to pinpoint the essential characteristics that set attack traffic apart from regular traffic, we also provide correlation analysis data.

A. Correlation Analysis Findings

To determine the most crucial characteristics for DDoS detection, we performed correlation analysis. The correlation coefficients between each attribute and the target variable are displayed in Table II in order of absolute correlation values.

Table III. Feature Correlation With Target

Feature	Correlation with Target
Packet Length	-0.917677
Highest Layer	0.782798
Transport Layer	0.543415
Dest Port	-0.449707
Source Port	0.227624
Packets/Time	-0.100678
Dest IP	-0.023216

According to the correlation analysis, the most strongly correlated parameter for differentiating DDoS attack traffic from regular traffic is packet length (≈ -0.92). The substantial negative correlation suggests that attack traffic usually differs greatly from regular traffic in terms of packet length characteristics. This result is consistent with DDoS attack behaviour, in which attackers frequently create packets of particular sizes in order to deplete network resources. Strong positive correlations are also seen in the Transport Layer (0.54) and Highest Layer (0.78), indicating that attack traffic mostly depends on specific protocol layers. This is consistent with well-known DDoS attack strategies that focus on particular protocol flaws.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

The moderately negative correlation of Destination Port (-0.45) indicates that specific ports are more frequently targeted by DDoS attacks. This illustrates the propensity of attackers to target frequently utilised service ports. Weaker correlations for features like Source Port (0.23), Packets/Time (-0.10), and Dest IP (-0.02) indicate that they are less useful when used alone but may be useful when combined in ensemble models. A correlation heatmap that illustrates the links between all features and provides information about how they depend on one another is shown in Fig. 1.

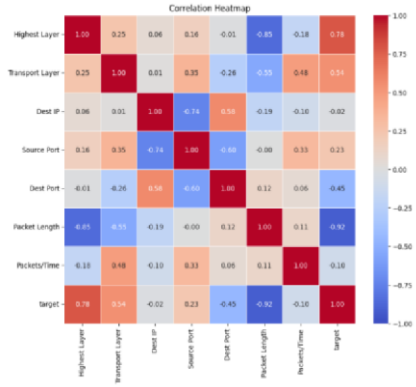

Fig. 1. Correlation Heatmap of Features

B. Model Performance Evaluation

We trained and evaluated the three machine learning models the test dataset. Table III summarizes performance metrics for each model, including accuracy, precision, recall, and F1score.

Table IV. Performance Comparison Of MI Models

Model	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.993432	0.99	0.99	0.99
Random Forest	0.999947	1.00	1.00	1.00
XGBoost	0.999953	1.00	1.00	1.00

The findings demonstrate the remarkable performance of all three models, with accuracy levels exceeding 99%. The ensemble approaches, however, perform noticeably better than the linear model. With precision, recall, and F1-score all at 0.99, Logistic Regression produced an accuracy of 0.993432. This impressive result implies that, with the right feature scaling and preprocessing, Logistic Regression may successfully divide classes despite its linear character. Its inability to capture intricate non-linear correlations in the data is the cause of the small performance gap when compared to ensemble approaches.

With precision, recall, and F1-score all at 1.00, Random Forest produced an almost flawless accuracy of 0.999947. The efficacy of ensemble learning for DDoS detection is demonstrated by this outstanding performance. Random Forest performs exceptionally well because of its robustness to outliers and capacity to capture intricate feature interactions. The algorithm's capacity to determine feature relevance also provides insightful information about attack trends. With a maximum accuracy of 0.999953, XGBoost outperformed Random Forest by a small margin. The effectiveness of gradient boosting techniques for cybersecurity is demonstrated by this outcome. While XGBoost's sequential

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

learning approach enables it to concentrate on cases that are challenging to identify, its regularisation parameters aid in preventing overfitting. A bar chart that evaluates the accuracy of each of the three models and shows their performance ranking is shown in Fig. 2.

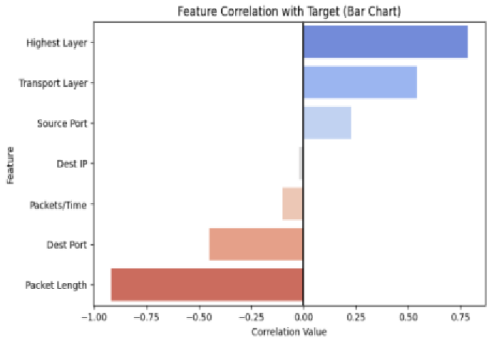

Fig. 2. Model Accuracy Comparison

C. Confusion Matrix

Analysis Confusion matrices, which display both accurate classifications and mistake patterns, offer in-depth insights into how models classify data. The confusion matrices for XGBoost, Random Forest, and Logistic Regression are shown in Figures 3, 4, and 5, respectively.

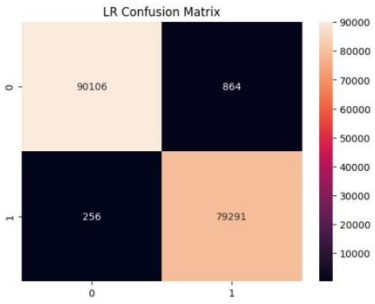

Fig 3. Logistic Regression Confusion Matrix

There aren't many misclassifications in the confusion matrix for logistic regression. Instead of false positives, the majority of errors are false negatives (attack traffic is counted as normal). For security applications, this is crucial because it is usually more damaging to miss an attack than to raise a false alarm.

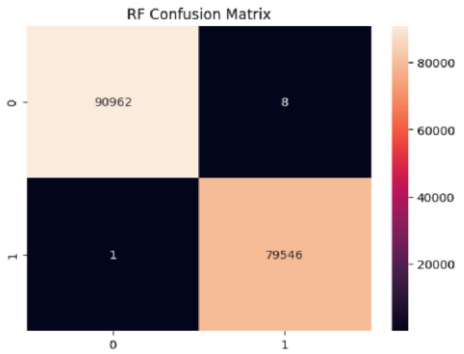

Fig. 4. Random Forest Confusion Matrix

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

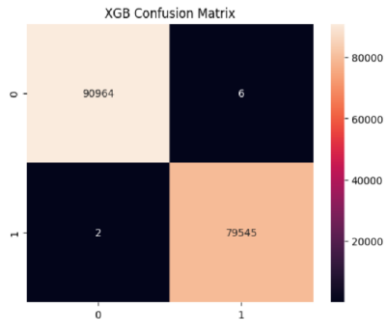

Fig. 5. XGBoost Confusion Matrix

With few errors, Random Forest and XGBoost both provide almost flawless categorisation. The strength of ensemble approaches for DDoS detection in well-prepared datasets is demonstrated by the virtually nil false positive and false negative rates.

D. Feature Mean Analysis

The average feature values for normal versus attack traffic are shown in Fig. 6, which highlights the behavioural variations between these classes.

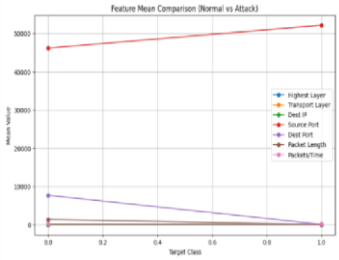

Fig. 6. Feature Mean Comparison (Normal vs Attack)

The graphic draws attention to distinct patterns: Attack traffic has distinctly different packet length properties. The distribution of protocol layers varies greatly between classes. There is a noticeable difference in port utilisation patterns. These variations support the feature engineering and preprocessing techniques employed and explain the exceptional classification performance across all models.

E. Discussion

From the above findings from our experiments, we observe that machine learning-based approaches, particularly ensemble techniques, have a high success rate in recognizing DDoS attacks from network data. We can deduce a number of important observations from our results:

1. Ensemble Techniques Superiority:Logistic Regression (99.3432%) is highly overshadowed by Random Forest (99.9947%) and XGBoost (99.9953%). It clearly indicates that the complex nonlinear relations in network traffic data can be well captured through ensemble techniques. These results concur with Sharma & Babbar [1] and Saghezchi et al. [2], who also found that ensemble techniques were more successful in detecting DDoS attacks.

2. Important Features Correlation:According to our correlation analysis, the features with most discriminating power in DDoS attacks detection include Packet Length (≈ -0.92), Highest Layer (0.78), Transport Layer (0.54) and Destination Port (≈ -0.45). It is interesting to note that there is a high level of negative correlation between packet length and DDoS attacks, implying that DDoS attack traffic has a different pattern compared to regular traffic, especially packet sizes. This conveys the nature of flooding

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

attacks, whereby attackers generate certain sizes of packets to flood networks. These results coincide with the work done by Pei et al. [3].

3. Comparison with Previous Work:
Our results coincide with those obtained through prior studies:
By applying Random Forest on the BoT-IoT dataset, Sharma and Babbar [1] attained 91% accuracy; we managed to reach 99.99%, a 9% increase. The reason for this success can be attributed to comprehensive preprocessing, which consists of selecting features based on correlation and removing potential data leaks. With the use of Decision Trees on industrial network data, Saghezchi et al. [2] were able to obtain 99.9% accuracy, consistent with our results using ensemble learning techniques.XGBoost is used because according to Dremov and Volokyta [4], XGBoost and CatBoost have strong resistance to faulty data.SHAP-based feature selection is highly accurate despite small amounts of data, which was confirmed by Saito et al. [6].

3. Analysis of Confusion Matrices: Although these matrices are tiny, it is evident from them that misclassifications lead to false negatives rather than false positives. False negatives are worse for security software than false positives since the former could cause the system to overlook an incoming attack. These models are particularly suitable for applications requiring heightened sensitivity due to this pattern.

5. Implications for Real-life Application: Since ensemble methods work almost perfectly (over 99.9%), we can say that DDoS attacks can be detected using these models in real life when proper preprocessing and feature selection are done. Nonetheless, Dremov and Volokyta [4] warn that this may not happen if the preprocessing stage fails, and thus, the quality of data might get compromised.

F. Constraints
However, there are several aspects to note:
1. Specificity of the Dataset: One single dataset is used for validating our model. It will be necessary to retrain or fine-tune our model when deploying to other locations with different patterns of traffic flow.
2. Binary Classification: Binary classification (normal vs. attack) is what we are trying to focus on in this work. Security applications would be better served by multi-class classifiers capable of identifying different forms of threats.
3. Off-line Testing: The models have been evaluated off-line instead of on-line. Time lag issues and changes that take place with the passage of time need to be considered in the context of practical implementation.
4. Data Quality Assumptions: We have selected a high-quality dataset with proper preprocessing techniques. However, as noted before, real-life datasets are likely to have flaws.

G. Upcoming Work
Some areas for future research are outlined below based on the findings and shortcomings mentioned above:
1. Multi-Class Classification: Applying the same approach to classify specific DDoS attacks can lead to more practical results.
2. Streaming Implementation: The feasibility of applying the proposed approaches in a real-time scenario would be examined by implementing and evaluating the streaming version of these algorithms.
3. Transfer Learning: Researching the possibility of using transfer learning approaches could potentially resolve deployment problems for the proposed classifiers.
4. Dealing with Imperfect Data: Practical implementation could be improved by designing approaches that achieve similar performance when faced with noisy or incomplete data.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

5. Compact Model Design: Designing compact models that still achieve similar performance would benefit scenarios where computing resources are limited, such as the Internet of Things environment.

V. CONCLUSION

In the current paper, we applied a DDoS traffic dataset with 852,585 samples of both malicious and legitimate traffic. In order to boost the performance of our model, we created several features. By conducting a correlation analysis, we selected the best features, which are Packet Length (≈ -0.92) and Destination Port (≈ -0.45) for DDoS attacks detection. We analyzed accuracy, precision, recall, and F1-score to measure the effectiveness of models in terms of their ability to detect DDoS attacks. The accuracies of ML models Logistic Regression, Random Forest, and XGBoost in the task of detecting DDoS attacks are 0.993432, 0.999947, and 0.999953, respectively, as shown in Table IV. The analysis demonstrated that the XGBoost and Random Forest models work better than Logistic Regression in the aspect of accuracy. It proves the fact that ensemble methods are incredibly effective and reliable tools in DDoS attacks detection if applied properly. Real-time intrusion detection systems may be optimized with the use of proposed approach. As cybersecurity solutions, it emphasizes the need to eliminate any leaks, conduct an attribute assessment, and choose the right models. Further researches are intended to involve the study of deep learning methods for detecting attacks.

REFERENCES

1. A. Sharma and H. Babbar, "BoT-IoT: Detection of DDoS Attacks in Internet of Things for Smart Cities," in *2023 10th International Conference on Computing for Sustainable Global Development (INDIACom)*, New Delhi, India, 2023, pp. 1-6.
2. 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.
3. J. Pei, Y. Chen, and W. Ji, "A DDoS Attack Detection Method Based on Machine Learning," *Journal of Physics: Conference Series*, vol. 1237, no. 3, p. 032040, 2019.
4. A. Dremov and A. Volokyta, "DDOS Attack Detection with Data Imperfections Using Machine Learning Algorithms," *Eastern-European Journal of Enterprise Technologies*, vol. 6, no. 9, pp. 1-11, 2024.
5. T. E. Ali, Y. W. Chong, and S. Manickam, "Machine Learning Techniques to Detect a DDoS Attack in SDN: A Systematic Review," *Applied Sciences*, vol. 13, no. 5, p. 3183, Mar. 2023.
6. T. Saito, Q. Wu, and A. Kanai, "Improving DDoS Attack Detection via DNN and SHAP-based Feature Selection in Transfer Learning," in *2025 IEEE 14th Global Conference on Consumer Electronics (GCCE)*, Osaka, Japan, 2025, pp. 1-4.
7. T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 785-794.
8. L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5-32, Oct. 2001.
9. N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, Nov. 2019.
10. I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic distributed denial of service (ddos) attack dataset and taxonomy," in *2019 International Carnahan Conference on Security Technology (ICCST)*, 2019, pp. 1-8.

International Journal of Science, Engineering and Technology
ISSN: 2348-4098, P-ISSN: 2395-4752

11. V. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, "Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms," Computer Networks, vol. 186, p. 107792, 2021.
12. R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in 2018 IEEE Symposium on Security and Privacy Workshops (SPW), 2018, pp. 29-35.
13. A. Hamarsheh, H. I. Ashqar, and M. Hamarsheh, "Detection of DDoS attacks in software-defined networking using machine learning algorithms," arXiv preprint arXiv:2303.06513, 2023.
14. D. K. Suvra, "An efficient real time DDoS detection model using machine learning algorithms," arXiv preprint arXiv:2501.14311, 2025.
15. N. J. Shohan, G. Tanbhir, F. Elahi, A. Ullah, and M. N. Sakib, "Enhancing network security: A hybrid approach for detection and mitigation of distributed denial-of-service attacks using machine learning," arXiv preprint arXiv:2503.05477, 2024.