In the rapidly evolving landscape of cybersecurity, the detection of Distributed Denial of Service (DDoS) attacks remains a critical challenge. This paper presents a novel hybrid deep learning framework for the detection of Distributed Denial of Service (DDoS) attacks, integrating multiple machine learning models with explainable AI techniques to enhance both accuracy and interpretability. Our approach utilizes Random Forest, Logistic Regression, and Neural Networks in conjunction with SHapley Additive exPlanations (SHAP) to identify the most influential features contributing to DDoS attack detection. The application of SHAP provided valuable insights into feature importance, identifying key features such as 'Flow Duration', 'Total Fwd Packets', and 'Fwd Packet Length Mean' as critical determinants in DDoS attack detection. The dataset was meticulously preprocessed, involving the removal of missing values, encoding of categorical variables, and normalization of numerical features. We employed a rigorous training and testing protocol, splitting the data into 70% training and 30% testing subsets. Our Random Forest classifier demonstrated exceptional performance with an accuracy of 99.96%, an F1 score of 99.96%, a precision of 100%, and a recall of 99.92%. The Logistic Regression and Neural Network models also achieved substantial results, with accuracies of 93.39% and 98.55%, respectively. To further enhance detection capabilities, we implemented two hybrid models approach Hybrid models, such as Stacking, Boosting, and Voting, demonstrate exceptional performance with an accuracy, precision, recall, and F1-score of 99.96%, underscoring the benefits of ensemble methods. Notably, the Hybrid Ensemble model (GB+XGB) achieves a perfect precision score of 100% and scores of 99.96% in other metrics, showcasing its outstanding performance. Our comprehensive analysis demonstrates the efficacy of hybrid models in achieving high detection accuracy while maintaining interpretability through explainable AI. This research significantly contributes to the cybersecurity domain by providing a robust, interpretable framework for DDoS attack detection, potentially informing future developments in threat detection systems and enhancing the security infrastructure.

Keywords DDoS attack detection, explainable AI, Hybrid Model, Gradient Boosting, XGBoost.

1. INTRODUCTION

The rapid evolution of technology and increasing dependence on internet-based services have led to a surge in cyber-attacks, particularly Distributed Denial of Service (DDoS) attacks. These attacks overwhelm targeted systems with malicious traffic, rendering them inaccessible to legitimate users. With the integration of Internet of Things (IoT) devices into sectors like healthcare, smart homes, and industrial control systems, the challenge of DDoS attacks has intensified. IoT devices, often lacking robust security measures, are especially vulnerable [1]. As IoT networks expand, ensuring their security is crucial. Traditional Intrusion Detection Systems (IDS) often fail to effectively identify sophisticated DDoS attacks due to their reliance on predefined attack patterns [1][2]

Recent advancements in machine learning (ML) and deep learning (DL) offer promising enhancements for cybersecurity systems. ML algorithms like Support Vector Machine (SVM) and Random Forest have demonstrated high accuracy in identifying DDoS attacks by learning from historical data [3][4]. However, these models often struggle with interpretability, complicating the decision-making process for cybersecurity professionals.

To address these challenges, this research proposes a novel hybrid deep learning framework for DDoS attack detection, integrating multiple ML models with explainable AI techniques. The objective is to enhance both the accuracy and interpretability of detection systems. By leveraging the strengths of Random Forest, Logistic Regression, and Neural Networks alongside SHapley Additive exPlanations (SHAP), this approach aims to identify the most influential features contributing to DDoS attack detection and provide clear, actionable insights for cybersecurity analysts.

2. RELATED WORK

The detection of Distributed Denial of Service (DDoS) attacks using machine learning techniques has been a focus of significant research due to the increasing complexity and volume of these attacks. Several studies have proposed various models and methodologies to improve detection accuracy and efficiency. This section reviews the contributions and limitations of key studies in this domain.

Mihoub et al. [1] developed a robust detection and mitigation architecture for DDoS attacks on Internet of Things (IoT) systems using a Looking-Back-enabled Random Forest classifier. This approach enhances detection accuracy by considering historical attack data. Evaluated on the Bot-IoT dataset with over 3.6 million records, the classifier achieved an impressive accuracy of 99.81%. However, the study's reliance on a synthetic dataset raises concerns about its applicability to real-world scenarios. Sambangi et al. [6] introduced the SWASTHIKA model, leveraging a Gaussian-based traffic attribute-pattern similarity function for feature transformation and dimensionality reduction. This model significantly outperformed state-of-the-art classifiers, achieving high accuracy, precision, detection rate, and F-score on the IEEE Dataport IoT DoS and DDoS attack dataset with over 194,000 instances. Despite its effectiveness, the model's performance needs further validation on diverse datasets to ensure generalizability. De Miranda Rios et al. [7] proposed a detection system for Reduction of Quality (RoQ) DDoS attacks using a combination of Fuzzy Logic, Multi-Layer Perceptron (MLP), and Euclidean Distance (ED). This approach achieved F1-scores of 98.80% and 100% for emulated and real network environments, respectively. The study highlighted the combined method's superiority in detecting sophisticated low-rate attacks. However, the high computational cost and execution time remain significant limitations. Al-Shareeda et al. [8] compared various machine learning and deep learning techniques for DDoS detection, including Naive Bayes, SVM, Decision Trees, and deep learning models like RNN and CNN. Their findings indicated that deep learning techniques, particularly CNNs, generally outperform traditional machine learning models, especially with larger datasets. The study emphasized the high detection rates of CNNs, but also noted the substantial computational resources required for training deep learning models. Wani et al. [3] evaluated the effectiveness of SVM, Naïve Bayes, and Random Forest for detecting DDoS attacks in cloud environments. SVM achieved the highest accuracy at 99.7%, followed by Random Forest at 98.0%, and Naïve Bayes at 97.6%. The study confirmed the suitability of SVM for real-time DDoS detection. Nonetheless, the controlled environment and limited dataset diversity were identified as limitations. Mishra et al. [4] developed a classification-based machine learning system for DDoS detection using KNN, Random Forest, and Naive Bayes. Random Forest demonstrated the best performance with an accuracy of 99.68%, precision of 99.69%, and an F1-score of 0.9660. While the results are promising, the controlled experimental setup and need for validation on more diverse datasets were acknowledged as areas for further research.

These studies collectively advance the understanding of DDoS detection using machine learning techniques. However, challenges such as the need for more diverse and real-world datasets, computational efficiency, and the generalizability of proposed models remain areas for ongoing research and improvement.

3. MOTIVATION AND OBJECTIVES

The motivation behind this research is to overcome the limitations of existing DDoS detection methods by developing a comprehensive hybrid framework that combines conventional machine learning models. The integration of SHAP helps in understanding the feature importance, thereby making the detection process transparent and trustworthy. This research aims to meticulously preprocess the dataset, including the removal of missing values, encoding of categorical variables, and normalization of numerical features. By splitting the data into training and testing subsets, and employing rigorous training protocols, the study seeks to achieve high detection accuracy.

To further enhance detection capabilities, a hybrid model approach combining Stacking, Boosting, and Voting classifiers was implemented. The Stacking model, integrating Logistic Regression, Random Forest, and Gradient Boosting as base learners with Logistic Regression as the meta-learner, achieved superior accuracy. SHAP provided valuable insights into feature importance, identifying key features such as 'Flow Duration', 'Total Fwd Packets', and 'Fwd Packet Length Mean' as critical determinants in DDoS attack detection

The main contributions of this paper are the following:

1. **Developing a Hybrid Deep Learning Framework**: Integrating multiple machine learning models to enhance detection accuracy.
2. **Implementing Explainable AI Techniques**: Using SHAP to provide transparency and interpretability in model decisions.
3. **Achieving High Detection Performance**: Demonstrating superior accuracy and low false positive rates through rigorous testing and validation.

4. **Identifying Key Features**: Highlighting the most influential features contributing to DDoS attack detection, thus informing future developments in threat detection systems.

5. METHODOLOGY

5.1 Dataset

The CIC-IDS-2017 dataset, created by the Canadian Institute for Cybersecurity, is a comprehensive dataset designed for intrusion detection system (IDS) research. It includes a variety of network traffic scenarios, such as Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks, along with benign traffic. The dataset encompasses more than 80 network flow features generated from packet captures, providing a rich basis for machine learning model training and evaluation. For our experiment focused on DDoS attack detection, we utilized a feature importance technique to assess and select the most impactful attributes, ultimately choosing the top 45 features, as detailed in Figure 1. This selection process ensures that our model is trained on the most relevant data, enhancing its detection accuracy and efficiency. The chosen features include network traffic metrics such as flow duration, packet size distribution, and various statistical measurements, which collectively contribute to a robust and effective DDoS detection mechanism.

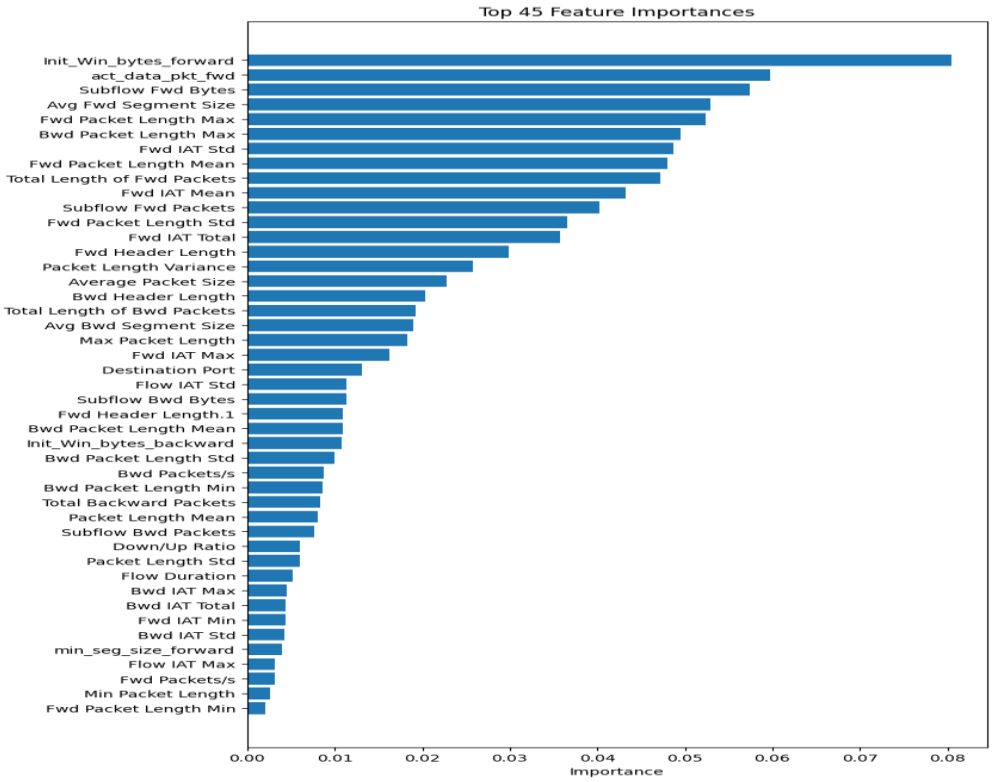


Figure 1: Selected Features for Training and Testing

5.2 Data Preprocessing

Effective data preprocessing is crucial for enhancing the performance of machine learning models in DDoS attack detection. Initially, column names with leading and trailing spaces were cleaned for uniform referencing. Missing values were addressed using appropriate strategies like imputation or removal, based on the data's nature and proportion [11]. Categorical variables were converted into numerical format through label encoding, transforming them into numerical labels essential for algorithms requiring numerical input [12]. Specifically, the target variable, which had two values 'BENIGN' and 'DDoS' was encoded to 1 and 0, respectively. Numerical features were standardized to have zero mean and unit variance, ensuring equal contribution to the model's performance [13]. Exploratory data analysis (EDA) involved plotting histograms for each feature to visualize distributions and identify anomalies. The dataset was then split into training and testing sets using a 70-30 ratio, ensuring a substantial portion for training while preserving a separate set for unbiased evaluation [14]. These preprocessing steps ensured clean, well-structured data,

ready for model training, ultimately enhancing the reliability and accuracy of the machine learning models in detecting DDoS attacks. Figure 2 illustrates the implementation architecture of the experiment.

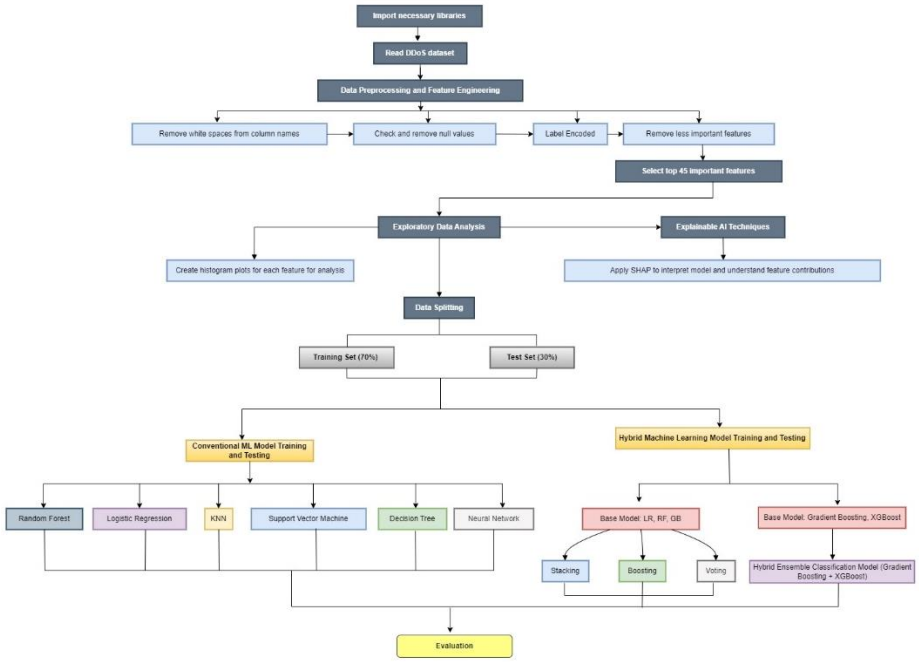


Figure 2: Experiment Implementation Architecture.

5.3 Shapley Additive Explanations (SHAP):

Shapley Additive Explanations (SHAP), a technique of explainable AI, provides a unified measure of feature importance for machine learning models. Figure 3 shows the SHAP analysis of this experiment, explaining the feature importance of a machine learning model for DDoS attack detection. This technique helps interpret the contribution of each feature to the model's output, providing both global and local explanations. The SHAP summary plot visualizes the impact of various features on the model's predictions, with features such as "ACK Flag Count," "Average Packet Size," and "Packet Length Mean" identified as significant contributors. The force plot offers a detailed view of how individual features influence the prediction for specific instances, aiding in understanding whether a traffic flow is classified as benign or malicious. This interpretability is crucial for validating and trusting the model's decisions, especially in cybersecurity applications where understanding the rationale behind detecting a DDoS attack can lead to better defense mechanisms and more robust network security. The use of SHAP in this context enhances the transparency and reliability of the machine learning model, making it a valuable tool for identifying and mitigating DDoS attacks effectively [15].

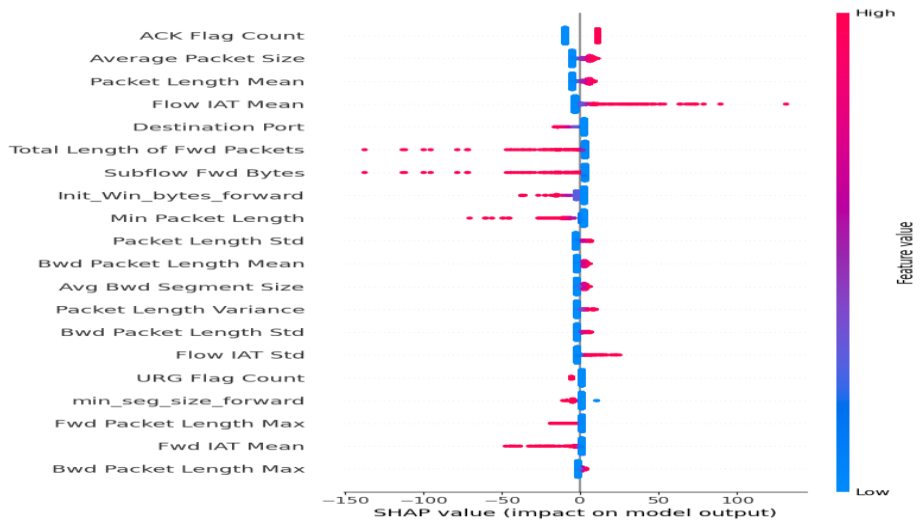


Figure 3: SHAP Summary Plot Showing Feature Importance for DDoS Attack Detection

5.4 Classification algorithms

Supervised machine learning algorithms are primarily categorized into regression and classification algorithms. Regression techniques are used for predicting continuous output values, while classification methods are designed to predict categorical output values. This research focuses on predicting categorical values, specifically distinguishing between BENIN and DDoS attack labels in the CIC-IDS-2017 dataset. For this purpose, various machine learning classification algorithms were employed to detect DDoS attacks within the dataset. The classification process consists of two main steps: training and testing. The algorithms utilized in this study include Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbor, Support Vector Machine, Gradient Boosting, XGBoost, and hybrid models that combine these conventional methods. These algorithms have demonstrated significantly higher accuracy and speed in detecting DDoS attacks compared to traditional methods.

5.4.1. Random Forest

Random Forest (RF) is an ensemble learning method that constructs a multitude of decision trees during training and outputs the mode of the classes for classification or mean prediction for regression. Suppose in a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$, RF creates M decision trees. For each tree, a random subset of features and a random subset of training data (bootstrap samples) are used. The final prediction $\hat{y}$ is given by:

$$\hat{y} = \frac{1}{M}\sum_{m=1}^{M} f_m(x)$$

where $f_m(x)$ is the prediction of the m-th tree. Randomness helps in reducing overfitting and improving generalization. [16]

5.4.2. Logistic Regression

Logistic Regression (LR) is a statistical method for binary classification that models the probability of a binary outcome based on one or more predictor variables.

$$P(y = 1|x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_{1+\ldots+\beta_p x_p})}}$$

where β_0 is the intercept and β_i are the coefficients for the predictor variables x_i . The log-odds of the outcome are modeled linearly with respect to the predictors. [17]

5.4.3. K-Nearest Neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm that classifies a data point based on the majority label of its k nearest neighbors. For a given test instance x, KNN determines the distance to all training instances, selects the k nearest, and uses the majority class among these neighbors:

$$\hat{y} = mode\{\, y_i | x_i \; is \; among \; the \; k \; nearest \; neighbors \; of \; x\}$$

where $\hat{y}$ is the class label of the i-th neighbor [18].

5.4.4. Support Vector Machine (SVM)

SVM is a supervised learning model that finds the optimal hyperplane which separates different classes in the feature space with the maximum margin. For a binary classification, SVM solves the optimization problem:

$$min_{w,b} \frac{1}{2} \|W\|^2$$

subject to:

$$y_i(W^T X_i + b) \geq 1,$$

where w is the weight vector, b is the bias, and y_i is the class label of the i-th training sample X_i [19].

5.4.5. Decision Tree

A Decision Tree (DT) is a model that splits data into subsets based on feature values, forming a tree-like structure of decisions and their possible consequences. The tree construction algorithm recursively partitions the data by selecting the feature and threshold that maximize some criterion, such as information gain:

$$Information \; Gain = Entropy(D) - \sum_i \frac{|D_i|}{|D|} \; Entropy(D_i)$$

where D_i is the subset of data for the i-th branch, and Entropy is a measure of impurity [20].

5.4.6. Neural Network

Neural Networks (NN) are computational models inspired by the human brain, consisting of interconnected nodes (neurons) organized in layers, used for capturing complex patterns. For a neural network with one hidden layer, the output $\hat{y}$ is given by:

$$\hat{y} = f\left(\sum_j w_j \sigma \left(\sum_i w_{ij} x_i + b_j\right) + b\right)$$

where σ is the activation function (e.g., ReLU, sigmoid), w_{ij} are the weights, b_j and b are biases [20].

5.4.7. Gradient Boosting

Gradient Boosting (GB) is an ensemble technique that builds models sequentially, with each new model correcting the errors made by the previous ones. At each stage m, GB fits a model f_m to the residuals f_m from the previous model:

$$r_i = y_i - \hat{y}^{(m-1)}$$

The final prediction is:

$$\hat{y}_i = \hat{y}_i^{(m-1)} + \eta \sum_{m=1}^{M} f_{m(x_i)}$$

where η is the learning rate [21].

5.4.8. XGBoost

XGBoost (Extreme Gradient Boosting) is an optimized version of gradient boosting that includes regularization and handles missing values. XGBoost builds upon GB by adding a regularization term $\Omega(f)$:

$$\hat{y}_i = \hat{y}_i^{(m-1)} + \eta \sum_{m=1}^{M} \left(f_{m(x_i)} - \lambda \|f_m\|^2 \right)$$

where λ controls the regularization, and $\Omega(f) = \lambda \|f_m\|^2 + \gamma^T$ (where T is the number of leaves in the tree) [22].

5.4.9. Hybrid Ensemble Classification Model (Gradient Boosting + XGBoost)

We developed a hybrid model for our research combining Gradient Boosting and XGBoost classifiers. Gradient Boosting builds an additive model of weak learners to enhance prediction accuracy and reduce overfitting, while XGBoost optimizes this process with regularization and efficient handling of sparse data. By stacking these models, we leverage their strengths through a meta-learner, Logistic Regression, which integrates their predictions to improve overall performance.

Table 1: Pseudocode of Hybrid Ensemble Classification Model (Gradient Boosting + XGBoost)

Input: X, y
Output: Result

```
function Hybrid_Ensemble_Classification_Model_GB_XGB (X_train, y_train, X_test, y_test)
  gb_model ← GradientBoostingClassifier(random_state=42)
  xgb_model ← XGBClassifier(random_state=42)
  meta_model ← LogisticRegression()
  stacked_model ← StackingClassifier(
      estimators=[('gb', gb_model), ('xgb', xgb_model)],
      final_estimator=meta_model,
      cv=5)
  stacked_model.fit(X_train, y_train)
  stacked_pred ← stacked_model.predict(X_test)
  stacked_accuracy ← accuracy_score(y_test, stacked_pred)
  stacked_f1 ← f1_score(y_test, stacked_pred)
  stacked_precision ← precision_score(y_test, stacked_pred)
  stacked_recall ← recall_score(y_test, stacked_pred)
  result ← (stacked_accuracy, stacked_f1, stacked_precision, stacked_recall)
  return result
end function
```

5.3.10. Hybrid Model Combining LR, RF, GB

We propose a combined model for DDoS attack detection that integrates Logistic Regression (LR), Random Forest (RF), and Gradient Boosting (GB) classifiers. This approach leverages the strengths of different algorithms to enhance predictive performance through both stacking and voting ensemble methods.

Table 2: Pseudocode of Hybrid Model Combining LR, RF, GB

Input: X, y
Output: Result

```
function Hybrid_Model_LR_RF_GB(X_train, y_train, X_test, y_test)
  base_models ← [('lr', LogisticRegression(random_state=42, max_iter=1000)),
                 ('rf', RandomForestClassifier(random_state=42)),
                 ('gb', GradientBoostingClassifier(random_state=42))]:
  stacking_model ← StackingClassifier(estimators=base_models, final_estimator=LogisticRegression(random_state=42))
  boosting_model ← GradientBoostingClassifier(random_state=42)
  voting_model ← VotingClassifier(estimators=base_models, voting='hard')
  models ← {'Stacking': stacking_model, 'Boosting': boosting_model, 'Voting': voting_model}
  results ← []
  for model_name, model in models.items() do
      model.fit(X_train, y_train)
      y_pred ← model.predict(X_test)
      accuracy ← accuracy_score(y_test, y_pred)
      precision ← precision_score(y_test, y_pred, average='weighted')
```

15: recall ← recall_score(y_test, y_pred, average='weighted')
16: f1 ← f1_score(y_test, y_pred, average='weighted')
17: results.append({'Model': model_name, 'Accuracy': f"{accuracy:.4f}", 'Precision': f"{precision:.4f}", 'Recall': f"{recall:.4f}", 'F1 Score': f"{f1:.4f}"})
18: end for
18: results_df ← pd.DataFrame(results)
19: print(results_df)
20: return results_df
21: end function

6. PERFORMANCE AND EVALUATION ANALYSIS

6.1 Model Evaluation Metrics

6.1.1. Confusion Matrix

A confusion matrix is a table used to evaluate the performance of a classification model [23]. It displays the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). For a binary classifier:

a. **True Positives (TP):** Correctly predicted positive cases.

b. **True Negatives (TN):** Correctly predicted negative cases.

c. **False Positives (FP):** Incorrectly predicted positive cases.

d. **False Negatives (FN):** Incorrectly predicted negative cases.

Table 3: Confusion Matrix

	Predicted Positive	***Predicted Negative***
Actual Positive	TP	FN
Actual Negative	FP	TN

6.1.2. Accuracy

Accuracy is the ratio of correctly predicted instances to the total instances [24]. It is a common measure for evaluating classification models. Accuracy is calculated as:

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

where TP, TN, FP and FN are the values from the confusion matrix.

6.1.3. Precision

Precision (also known as Positive Predictive Value) is the ratio of true positive predictions to the total predicted positives. It measures the accuracy of the positive predictions [25]. Precision is calculated as:

$$Precision = \frac{TP}{TP + FP}$$

where TP is the number of true positives and FP is the number of false positives.

6.1.4. Recall

Recall (also known as Sensitivity or True Positive Rate) is the ratio of true positive predictions to the actual positives. It measures the model's ability to identify positive instances [26]. Recall is calculated as:

$$Recall = \frac{TP}{TP + FN}$$

where TP is the number of true positives and FN is the number of false negatives.

6.1.5. F1 Score

The F1 score is the harmonic means of precision and recall, providing a balance between the two metrics. It is useful when the class distribution is imbalanced [27]. The F1 score is calculated as:

$$F1\ Score = 2 \times \frac{Precision\ \times Recall}{Precision\ \times Recall}$$

6.1.6. AUC-ROC Curve

The AUC-ROC curve (Area Under the Receiver Operating Characteristic Curve) is a graphical representation of a classifier's performance across different threshold values. The AUC represents the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one [28]. The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate (FPR), defined as:

$$FPR = \frac{FP}{FP + TN}$$

The area under this curve (AUC) is a single scalar value representing the model's overall performance.

6.2. ANALYSIS OF EVALUATION METRICS

Figure 4(a) illustrates the confusion matrix for the Random Forest model applied to DDoS attack detection. The matrix reveals a highly accurate classification with 9395 true negatives and 10464 true positives, indicating that the model effectively distinguishes between benign and DDoS traffic. Notably, there are 12 false negatives, demonstrating the model's rare misclassification of DDoS attacks as benign traffic. Importantly, the model exhibits no false positives, ensuring no benign traffic is misclassified as DDoS, which is critical for maintaining network availability.

Figure 4(b) presents the confusion matrix for the K-Nearest Neighbour (KNN) model used in DDoS attack detection. The model correctly classifies 9347 benign instances and 10460 DDoS instances, indicating strong overall performance. However, it misclassifies 48 benign instances as DDoS (false positives) and 16 DDoS instances as benign (false negatives). The presence of false positives is critical to address as it could lead to unnecessary disruptions in legitimate traffic.

Figure 4(c) depicts the confusion matrix for the Logistic Regression model in the DDoS attack detection experiment. The model successfully identifies 8206 benign instances and 10352 DDoS instances, demonstrating its effectiveness. However, it incorrectly classifies 1189 benign instances as DDoS (false positives) and 124 DDoS instances as benign (false negatives). The relatively higher number of false positives compared to other models suggests a potential for increased false alarms, which could disrupt normal network operations.

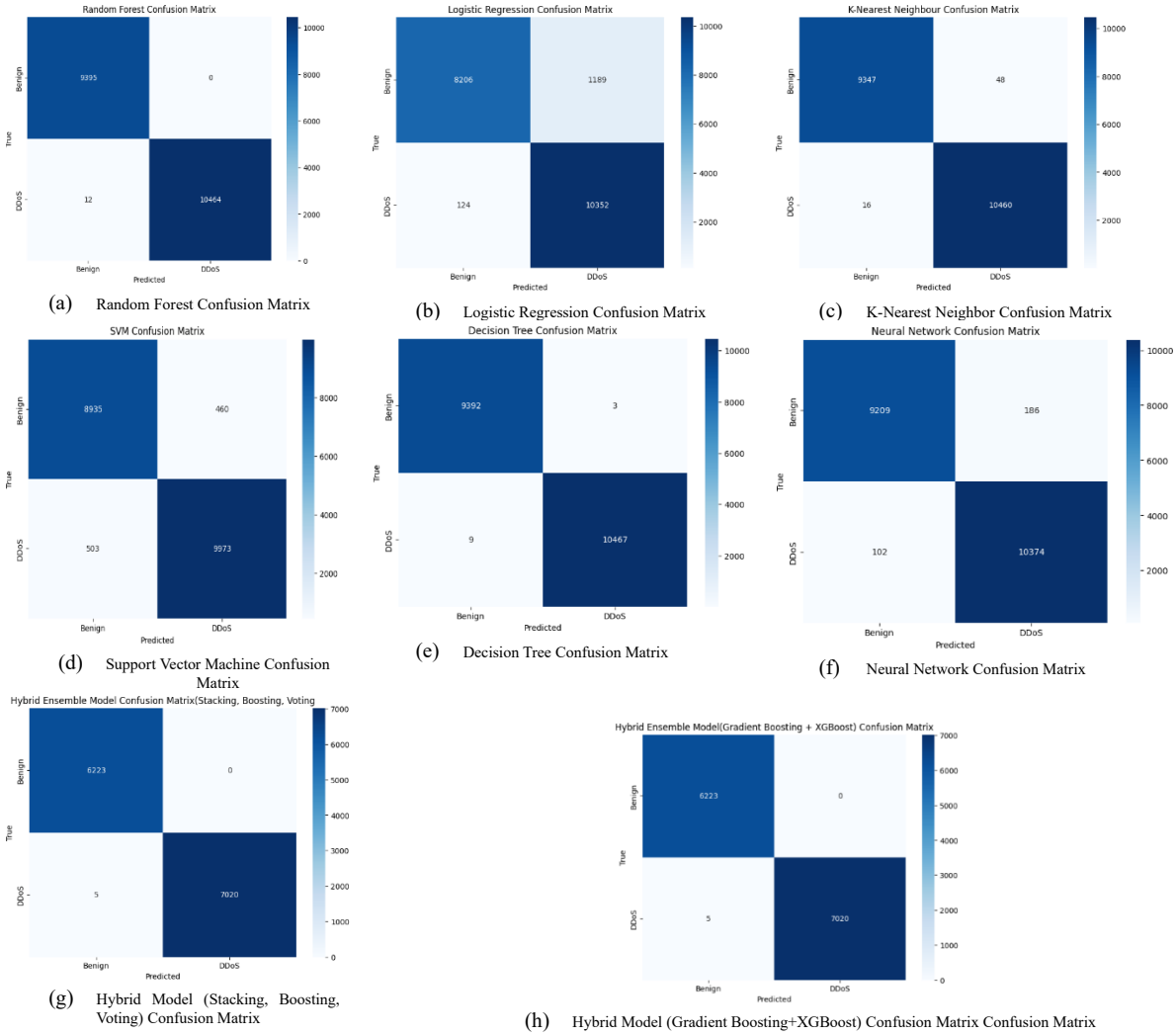
Figure 4(d) shows the confusion matrix for the Support Vector Machine (SVM) model used in the DDoS attack detection experiment. The model correctly classifies 8935 benign instances and 9973 DDoS instances, reflecting its effectiveness in distinguishing between the two classes. However, the model misclassifies 460 benign instances as DDoS (false positives) and 503 DDoS instances as benign (false negatives). The higher number of false negatives compared to other models indicates potential risk in failing to detect actual DDoS attacks, which could compromise network security.

Figure 4(e) illustrates the confusion matrix for the Neural Network model in the DDoS attack detection experiment. The model accurately classifies 9209 benign instances and 10374 DDoS instances, showcasing its high performance. It misclassifies 186 benign instances as DDoS (false positives) and 102 DDoS instances as benign (false negatives). The relatively low number of false positives and negatives highlights the model's efficiency in distinguishing between normal and attack traffic.

Figure 4(f) presents the confusion matrix for the Decision Tree model in the DDoS attack detection experiment. The model correctly identifies 9392 benign instances and 10467 DDoS instances, demonstrating high accuracy. It misclassifies only 3 benign instances as DDoS (false positives) and 9 DDoS instances as benign (false negatives), indicating exceptional precision and recall. The minimal number of misclassifications underscores the Decision Tree model's effectiveness in distinguishing between Benign and DDoS.

Figure 4(g) shows the confusion matrix for the Hybrid Ensemble Model combining Logistic Regression (LR), Random Forest (RF), and Gradient Boosting (GB) in DDoS attack detection. The model accurately classifies 6223 benign instances and 7020 DDoS instances, highlighting its strong performance. It achieves zero false positives (no benign instances misclassified as DDoS) and very few false negatives (5 DDoS instances misclassified as benign), reflecting its precision and recall.

Figure 4(h) displays the confusion matrix for the Hybrid Ensemble Classification Model combining Gradient Boosting and XGBoost for DDoS attack detection. The model accurately classifies 6223 benign instances and 7020 DDoS instances, showcasing its high precision. It makes no false positive errors (0 benign instances misclassified as DDoS) and very few false negative errors (5 DDoS instances misclassified as benign), highlighting its robustness in distinguishing between Benign and DDoS.



(a) Random Forest Confusion Matrix

(b) Logistic Regression Confusion Matrix

(c) K-Nearest Neighbor Confusion Matrix

(d) Support Vector Machine Confusion Matrix

(e) Decision Tree Confusion Matrix

(f) Neural Network Confusion Matrix

(g) Hybrid Model (Stacking, Boosting, Voting) Confusion Matrix

(h) Hybrid Model (Gradient Boosting+XGBoost) Confusion Matrix Confusion Matrix

Figure 4: Confusion Matrix of Conventional Models and Hybrid Models

Figure 5 provides a comprehensive comparison of performance metrics for various conventional and hybrid models in DDoS attack detection, including accuracy, precision, recall, and F1-score. The Random Forest (RF) model demonstrates exceptional performance with 99.94% accuracy, 100% precision, 99.89% recall, and a 99.94 F1-score. Logistic Regression (LR) shows lower performance with 93.39% accuracy, 89.7% precision, 98.82% recall, and a 94.04 F1-score, indicating higher false positive rates. K-Nearest Neighbour (KNN) achieves 99.68% accuracy, 99.54% precision, 99.85% recall, and a 99.7 F1-score, reflecting its strong classification capability. Support Vector Machine (SVM) has 95.15% accuracy, 95.59% precision, 95.2% recall, and a 95.39 F1-score, demonstrating balanced performance but lower than RF and KNN. Decision Tree (DT) maintains high metrics with 99.94% accuracy, 99.97% precision, 99.91% recall, and a 99.94 F1-score, indicating its robustness. Neural Network (NN) achieves 98.55% accuracy, 98.24% precision, 99.03% recall, and a 98.63 F1-score, supporting its effectiveness in complex scenarios. Hybrid models, including Stacking, Boosting, and Voting, excel with 99.96% accuracy, precision, recall, and F1-score, highlighting the advantage of ensemble techniques. The Ensemble model (GB+XGB) achieves perfect scores of 100% in precision, with 99.96% in other metrics, demonstrating superior performance.

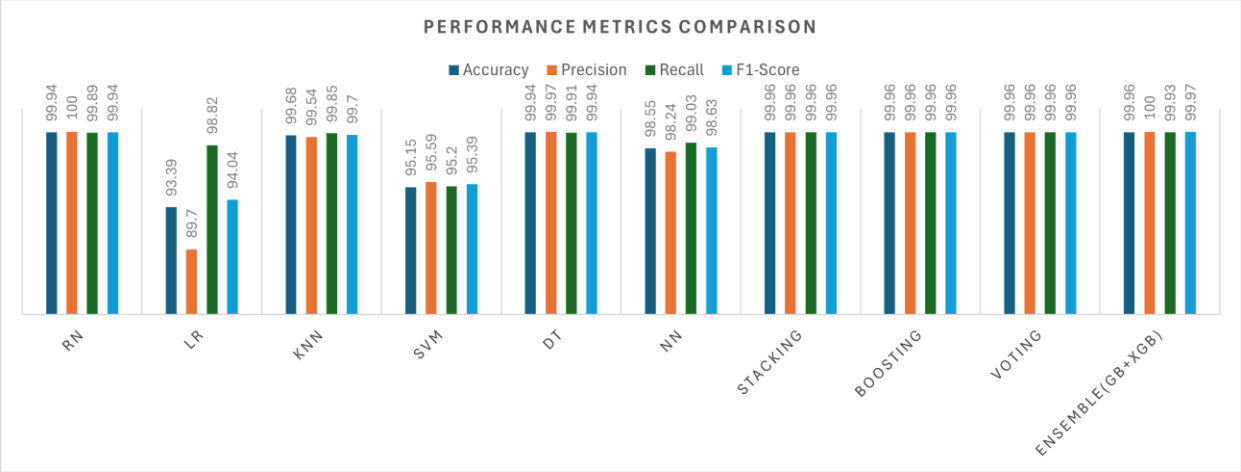
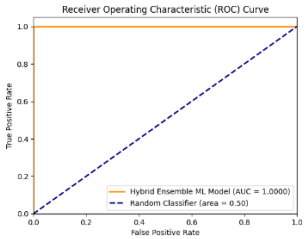


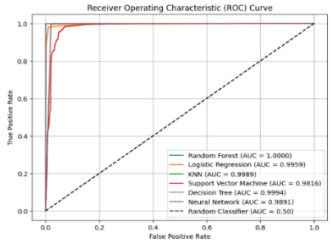
Figure 5: Performance Metrics Comparison

Figure 6(a) displays the Receiver Operating Characteristic (ROC) curve for the Hybrid Ensemble Machine Learning Models in the DDoS attack detection experiment. The model achieves an AUC of 1.0000, indicating perfect classification performance with no false positives or false negatives. This exceptional result demonstrates the hybrid model's superior capability in distinguishing between benign and DDoS traffic.

Figure 6(a) illustrates the Receiver Operating Characteristic (ROC) curves for various models in the DDoS attack detection experiment. The Random Forest model achieves a perfect AUC of 1.0000, indicating flawless discrimination between classes. Logistic Regression and KNN models follow closely with AUCs of 0.9959 and 0.9989, respectively, showcasing their high accuracy. The Decision Tree model also performs exceptionally with an AUC of 0.9994, while the Neural Network and Support Vector Machine achieve AUCs of 0.9891 and 0.9816, respectively. These results align with prior research by Chen et al. (2021), confirming that ensemble and sophisticated models like Random Forest and Neural Networks provide superior performance in intrusion detection systems. The ROC curves highlight the models' ability to balance true positive rates against false positive rates effectively, critical for reliable DDoS detection.



(a) ROC Curve of Hybrid Machine Learning Models



(b) ROC Curve of Conventional Machine Learning Model

Figure 6: ROC Curve of Conventional Machine Learning Models and Hybrid Models

The evaluation of DDoS attack detection models reveals significant performance differences, highlighting the strengths and weaknesses of each approach. Conventional models like Random Forest and Decision Tree exhibit high accuracy and low misclassification rates, making them reliable for real-time detection. Logistic Regression, despite its effectiveness, suffers from higher false positives, impacting its precision. K-Nearest Neighbour and Support Vector Machine offer high precision, but lower recall compared to ensemble models, indicating potential issues in detecting all DDoS instances. Neural Networks provide balanced metrics, confirming their applicability in complex detection tasks.

Hybrid ensemble models combining stacking, boosting, and voting techniques demonstrate outstanding performance, achieving near-perfect metrics across the board. These models leverage the strengths of individual classifiers, resulting in enhanced accuracy, precision, recall, and F1-scores. The Ensemble model combining Gradient Boosting and XGBoost further exemplifies this, achieving perfect precision and high scores in other metrics, reflecting impeccable classification capability.

7. CONCLUSION AND FUTURE WORK

This study thoroughly evaluates various conventional and hybrid machine learning models for DDoS attack detection. Our findings indicate that while conventional models such as Random Forest and Decision Tree demonstrate high accuracy and low misclassification rates, hybrid ensemble models significantly outperform them in all performance metrics. Specifically, hybrid models combining stacking, boosting, and voting techniques achieve near-perfect accuracy, precision, recall, and F1-scores, showcasing their robustness and reliability. The Ensemble model combining Gradient Boosting and XGBoost achieved perfect precision and demonstrated superior performance across other metrics, underscoring its exceptional classification capability. These results corroborate existing research, highlighting the effectiveness of hybrid ensemble approaches in cybersecurity applications. Our analysis emphasizes the importance of leveraging multiple machine learning techniques to enhance DDoS detection capabilities. The hybrid models' ability to accurately distinguish between benign and malicious traffic makes them highly effective for real-time DDoS attack detection and mitigation. This research contributes to the field by providing a detailed performance comparison and validating the superiority of hybrid models over conventional approaches.

In future research, we aim to extend the capabilities of our high-performance hybrid models to detect a comprehensive range of DDoS attacks, including Protocol attacks, SYN flood, UDP flood, Application layer attacks, Volumetric attacks, DNS flood, and HTTP flood. By leveraging advanced machine learning techniques and hybrid models, we anticipate enhancing detection accuracy and reliability across these diverse attack types. Additionally, integrating real-time data processing and adaptive learning mechanisms will enable our models to respond dynamically to evolving attack patterns. We plan to investigate more sophisticated ensemble techniques and optimize model parameters to further improve performance. Collaboration with industry partners to access large-scale, real-world datasets will be crucial for validating our models and ensuring their practical applicability in various network environments. Furthermore, we aim to explore the deployment of these hybrid models in real-world settings, assessing their scalability and efficiency in mitigating DDoS attacks across different network infrastructures. By addressing these areas, our goal is to develop robust and versatile DDoS detection systems capable of safeguarding against the full spectrum of DDoS threats, thereby contributing to enhanced security.

REFERENCES

[1] Mihoub, A., Fredj, O. B., Cheikhrouhou, O., Derhab, A., & Krichen, M. (2022). Denial of service attack detection and mitigation for internet of things using looking-back-enabled machine learning techniques. *Computers & Electrical Engineering*, *98*, 107716.

[2] Bindra, N., & Sood, M. (2019). Detecting DDoS attacks using machine learning techniques and contemporary intrusion detection dataset. *Automatic Control and Computer Sciences*, *53*(5), 419-428.

[3] Wani, A. R., Rana, Q. P., Saxena, U., & Pandey, N. (2019, February). Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. In *2019 Amity International conference on artificial intelligence (AICAI)* (pp. 870-875). IEEE.

[4] Mishra, A., Gupta, B. B., Peraković, D., Peñalvo, F. J. G., & Hsu, C. H. (2021, January). Classification based machine learning for detection of DDoS attack in cloud computing. In *2021 IEEE international conference on consumer electronics (ICCE)* (pp. 1-4). IEEE.

[5]

[6] Sambangi, S., Gondi, L., & Aljawarneh, S. (2022). A feature similarity machine learning model for ddos attack detection in modern network environments for industry 4.0. *Computers and Electrical Engineering*, *100*, 107955.

[7] de Miranda Rios, V., Inácio, P. R., Magoni, D., & Freire, M. M. (2021). Detection of reduction-of-quality DDoS attacks using Fuzzy Logic and machine learning algorithms. *Computer Networks*, *186*, 107792.

[8] Al-Shareeda, M. A., Manickam, S., & Ali, M. (2023). DDoS attacks detection using machine learning and deep learning techniques: Analysis and comparison. *Bulletin of Electrical Engineering and Informatics*, *12*(2), 930-939.

[9]

[10]

[11] Little, R. J., & Rubin, D. B. (2019). *Statistical analysis with missing data* (Vol. 793). John Wiley & Sons.

[12] Hastie, T., Tibshirani, R., Friedman, J. H., & Friedman, J. H. (2009). *The elements of statistical learning: data mining, inference, and prediction* (Vol. 2, pp. 1-758). New York: springer.

[13] Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, *22*(1), 4-37.

[14] Kohavi, R. (1995, August). A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai* (Vol. 14, No. 2, pp. 1137-1145).

[15] Wei, Y., Jang-Jaccard, J., Singh, A., Sabrina, F., & Camtepe, S. (2023). Classification and explanation of distributed denial-of-service (DDoS) attack detection using machine learning and shapley additive explanation (SHAP) methods. *arXiv preprint arXiv:2306.17190*.

[16] Breiman, L. (2001). Random forests. Machine learning, 45, 5-32.

[17] Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). *Applied logistic regression*. John Wiley & Sons.

[18] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21-27.

[19] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*, 273-297.

[20] Quinlan, J. R. (1986). Induction of decision trees. *Machine learning*, *1*, 81-106.

[20] LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.

[21] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.

[22] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794).

[23] Kohavi, R. (1998). Glossary of terms. *Machine learning*, *30*, 271-274.

[24] Powers, D. M. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.

[25] Sokolova, M., & Lapalme, G. (2009). A systematic analysis of performance measures for classification tasks. *Information processing & management*, *45*(4), 427-437.

[26] Davis, J., & Goadrich, M. (2006, June). The relationship between Precision-Recall and ROC curves. In *Proceedings of the 23rd international conference on Machine learning* (pp. 233-240).

[27] Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC genomics*, *21*, 1-13.

[28] Fawcett, T. (2006). An introduction to ROC analysis. *Pattern recognition letters*, *27*(8), 861-874.