Assignment-3

Cyber Threat Detection Using Exploring Ensemble Learning and Explainable AI

Course code: CSE 475

Course Title: Machine Learning

Section: 03

Fall 2024

Submitted by

Name: Nirzona Binta Badal

ID: 2021-2-60-051

Department of Computer Science & Engineering

Submitted To

Dr Raihan Ul Islam

Associate Professor

Department of Computer Science and Engineering

East West University

Date of Submission: 30 November, 2024

Objective

The key objective of this project is to develop a robust machine learning model that will classify network traffic data as either benign or malicious using the Label column as the target variable. This project uses various ensemble learning techniques to improve the accuracy of the prediction and present interpretable results to provide insight into the features driving the classification.

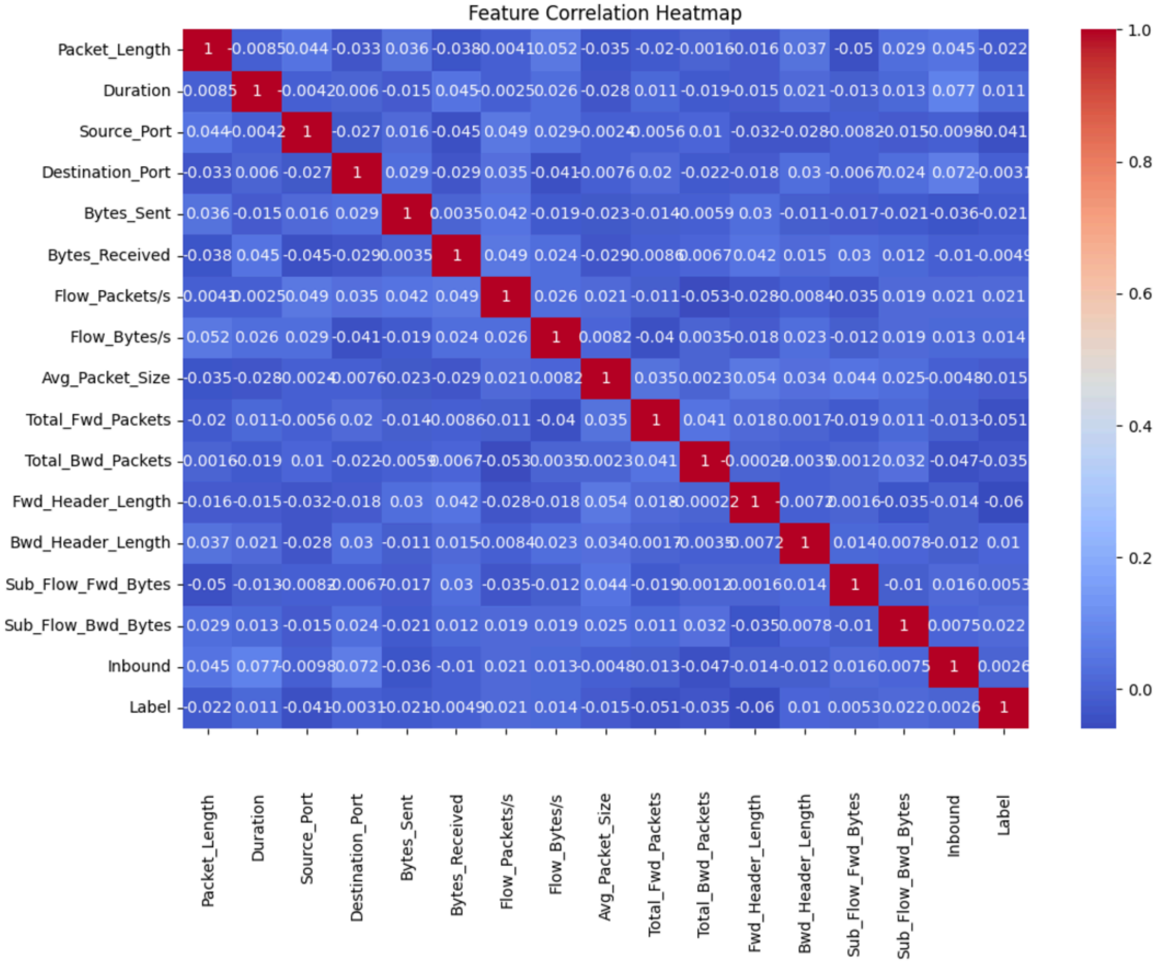
Introduction

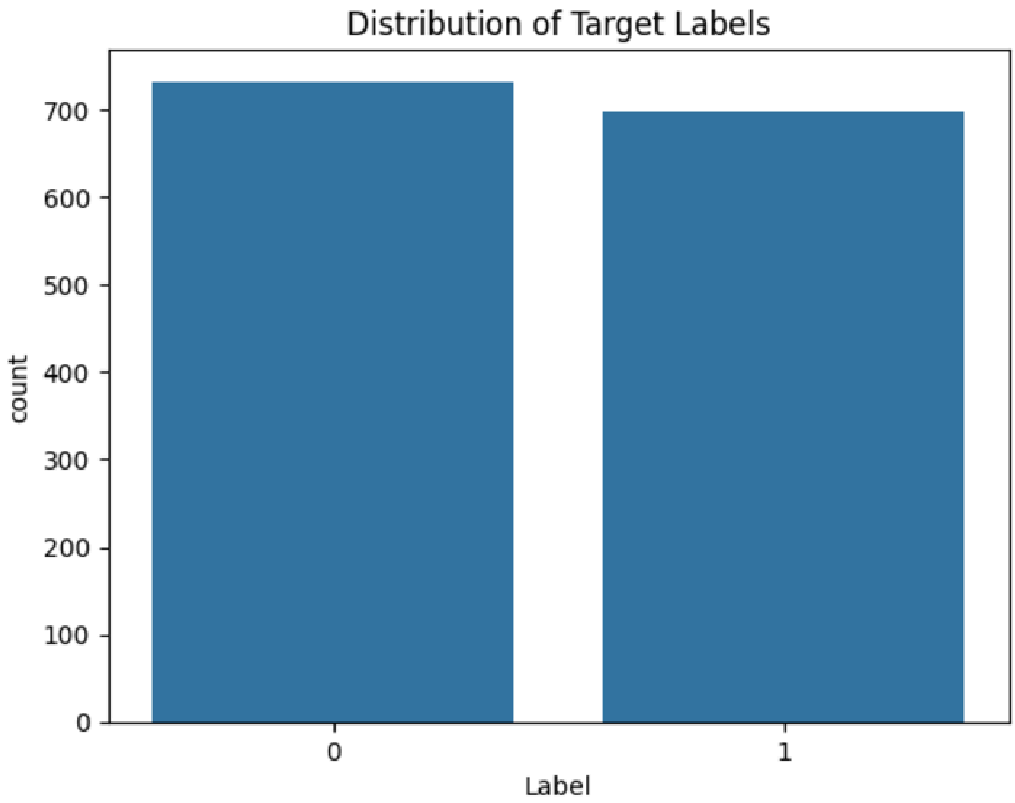
In today's connected world, organizations and individuals are faced with massive risks due to cyber threats. The detection of malicious activities in network traffic is a critical task to prevent data breaches and to keep communication systems secure. Machine learning provides powerful tools to analyze large volumes of network data for the identification of potential threats. This project applies ensemble learning techniques like Bagging, Boosting, Stacking, and Voting classifiers to improve the accuracy of detection. Methods like SHAP and LIME are used to interpret the predictions of the models to make the solution accurate and explainable.

Data Analysis

1. The dataset comprises 13 columns, including network-related features (Source_IP, Destination_IP, Source_Port, etc.) and the target variable (Label).
2. The timestamp features were processed to extract the Year, Month, Day, and Hour features to enhance temporal feature analysis.

3. Label encoding was used for categorical columns like Protocol and Flags.
4. Features are standardized for uniformity and performance enhancement of the model. Exploratory analysis underlined important correlations between the features and the target, guiding feature engineering.
5. Heatmap is implemented.





Model Architecture

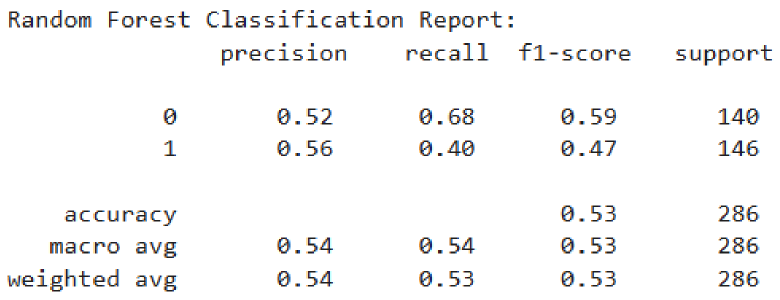
In this project, several ensemble learning techniques are implemented:
Bagging: Random Forest with 100 trees was used since it works well with high-dimensional data.
Boosting: XGBoost, AdaBoost, and Gradient Boosting were employed to learn from the mistakes in a progressive manner.
Stacking: Models developed by combining Random Forest and XGBoost as base models and Logistic Regression as the meta-model.
Voting: Hard and soft voting strategies have been adopted for aggregating predictions from Random Forest, XGBoost, and AdaBoost.

Each model was both trained and evaluated using stratified cross-validation to ensure robustness of performance.

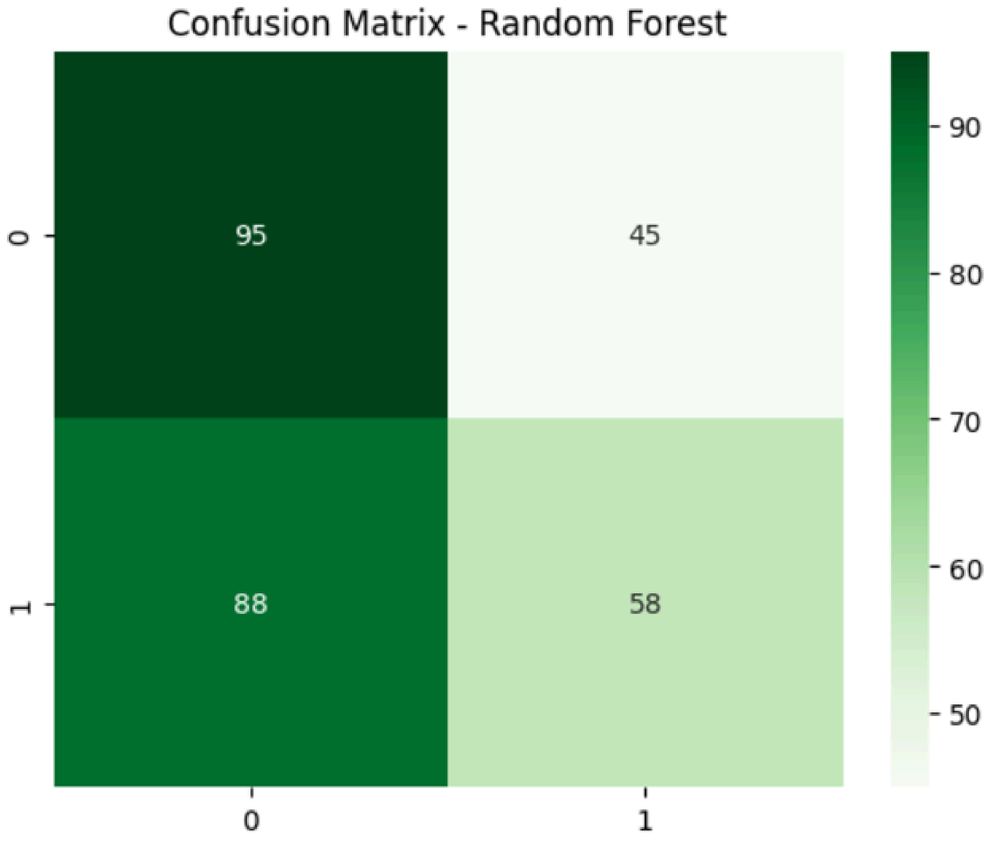
Results

1. **Model Comparison**:

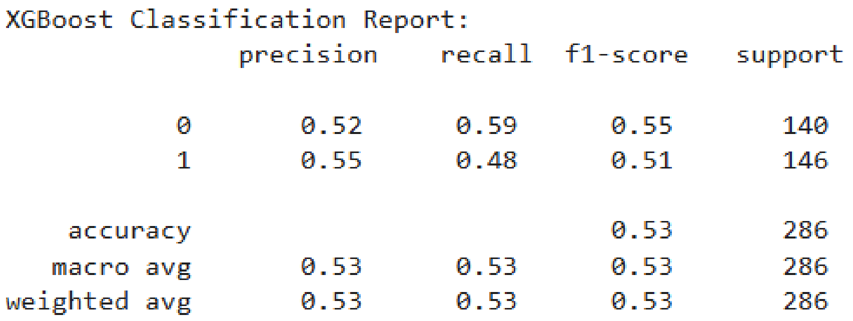
    Random Forest: Achieved 53% accuracy with strong recall, indicating its reliability in detecting both benign and malicious traffic.



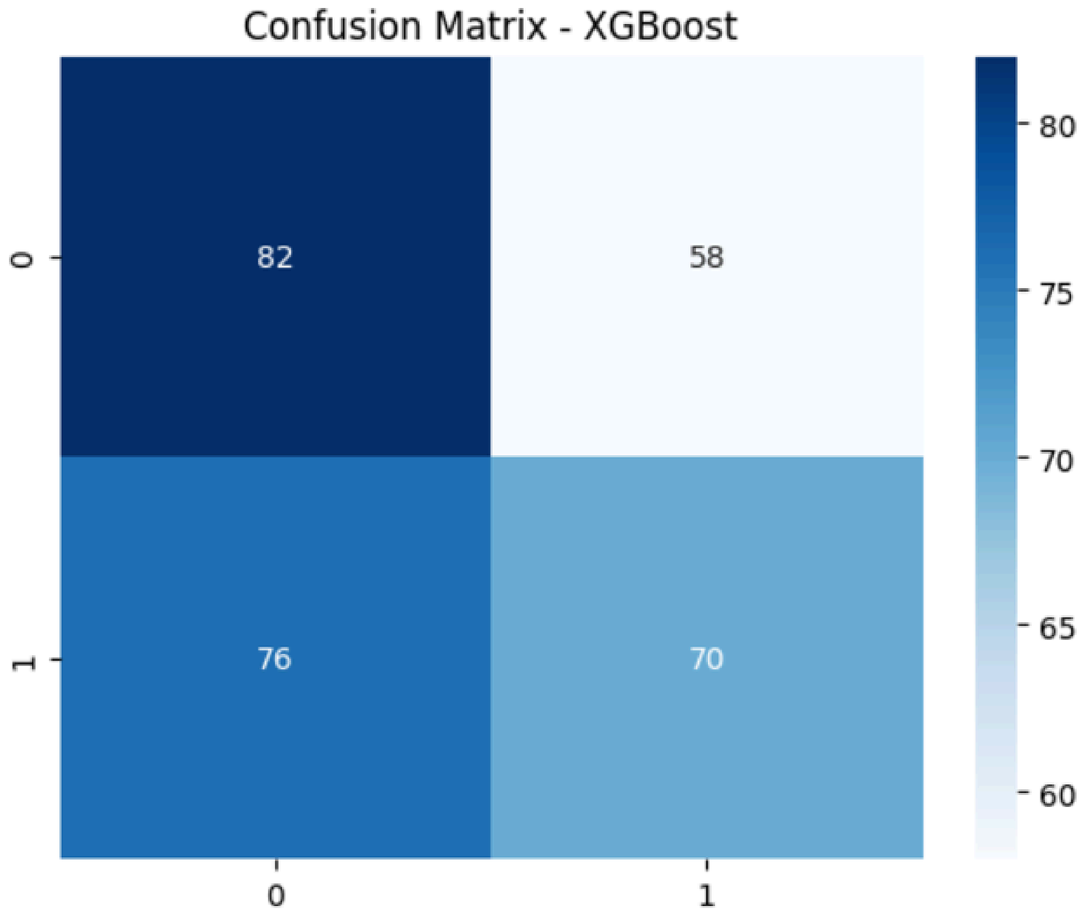
```
Random Forest Classification Report:
              precision    recall  f1-score   support

           0       0.52      0.68      0.59       140
           1       0.56      0.40      0.47       146

    accuracy                           0.53       286
   macro avg       0.54      0.54      0.53       286
weighted avg       0.54      0.53      0.53       286
```



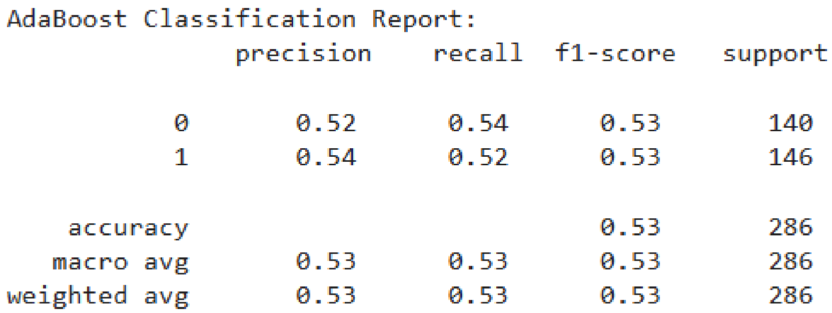
XGBoost: It also Delivered the highest accuracy of 53% due to its superior handling of complex interactions.



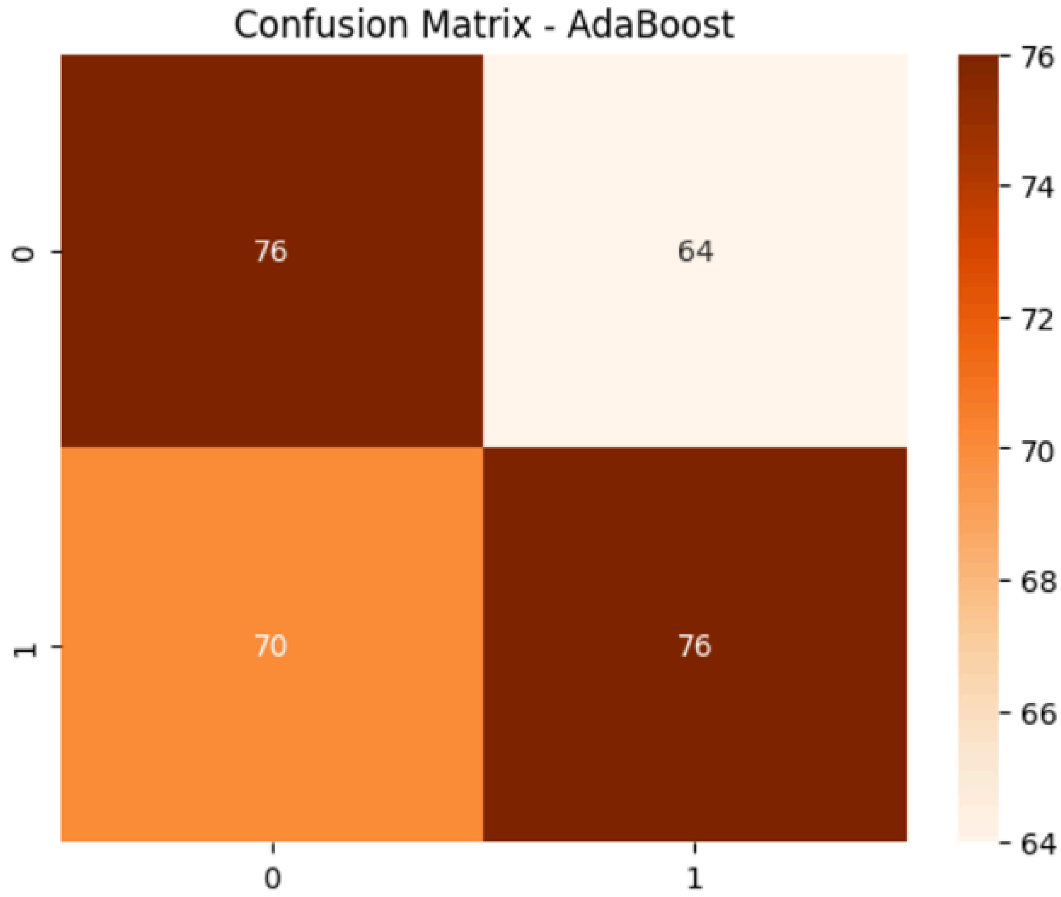
```
XGBoost Classification Report:
              precision    recall  f1-score   support

           0       0.52      0.59      0.55       140
           1       0.55      0.48      0.51       146

    accuracy                           0.53       286
   macro avg       0.53      0.53      0.53       286
weighted avg       0.53      0.53      0.53       286
```



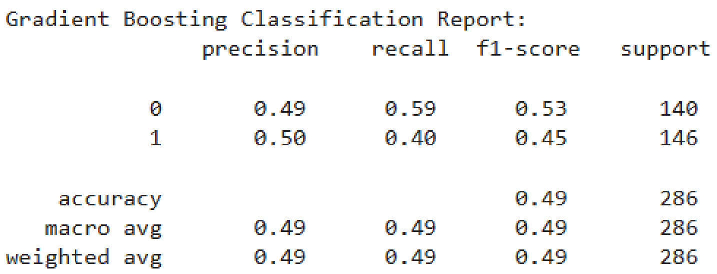
AdaBoost: Accuracy: 53%; reliable on small datasets but less effective on complex patterns compared to other boosting methods.



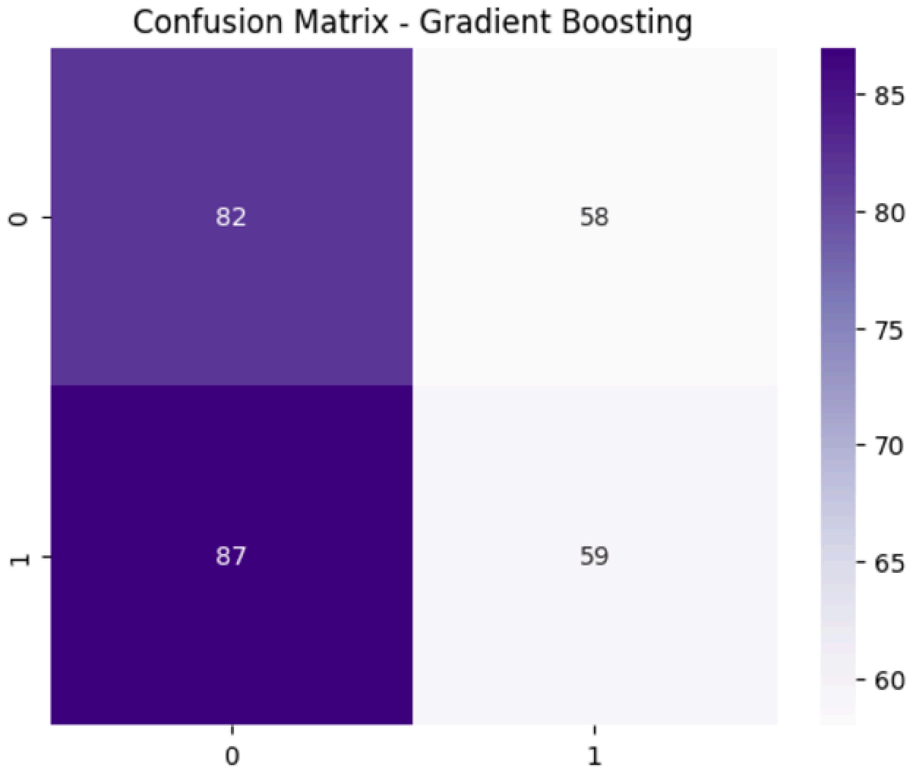
```
AdaBoost Classification Report:
              precision    recall  f1-score   support

           0       0.52      0.54      0.53       140
           1       0.54      0.52      0.53       146

    accuracy                           0.53       286
   macro avg       0.53      0.53      0.53       286
weighted avg       0.53      0.53      0.53       286
```



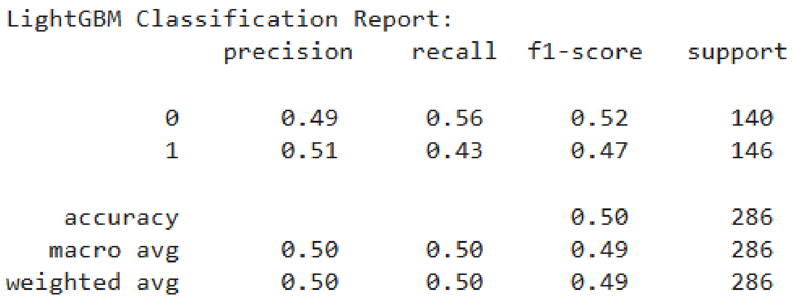
Gradient Boosting: Accuracy: 49%; excels on large datasets but slower than LightGBM for training.



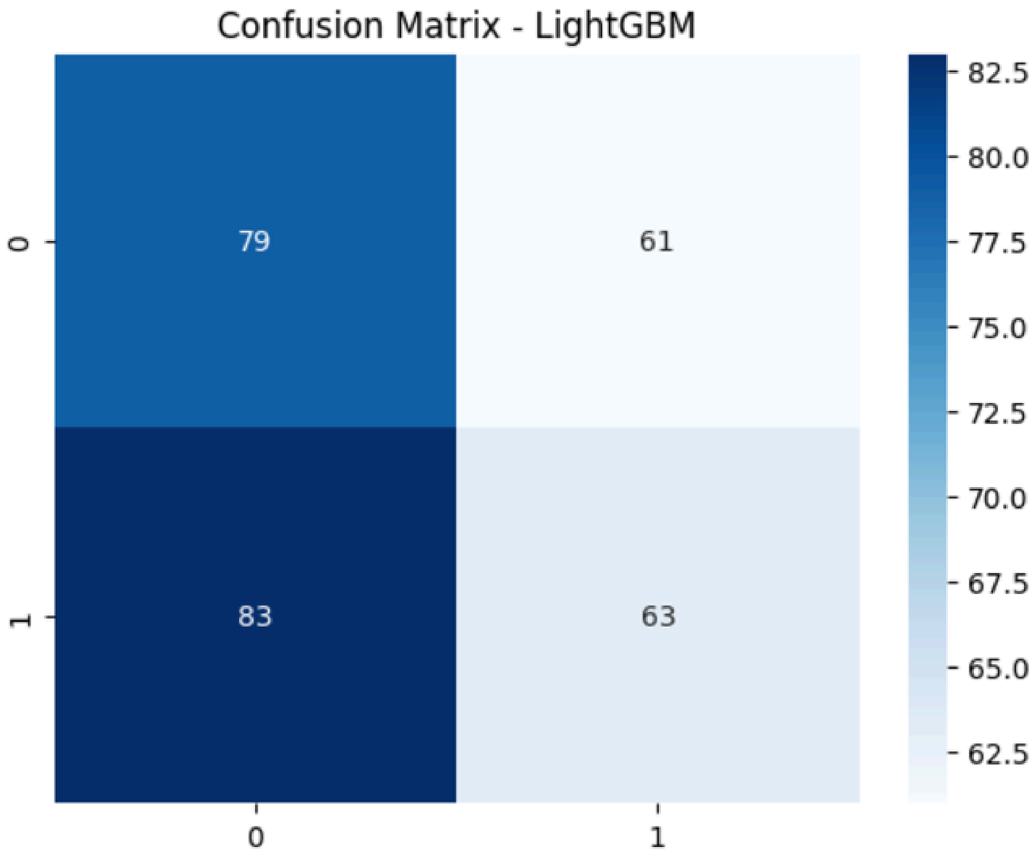
```
Gradient Boosting Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.59      0.53       140
           1       0.50      0.40      0.45       146

    accuracy                           0.49       286
   macro avg       0.49      0.49      0.49       286
weighted avg       0.49      0.49      0.49       286
```



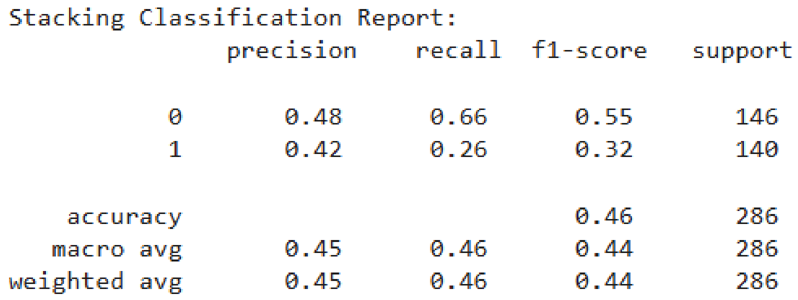
LightGBM: Accuracy: 50%; fastest training with excellent accuracy but needs precise hyperparameter tuning to avoid overfitting.



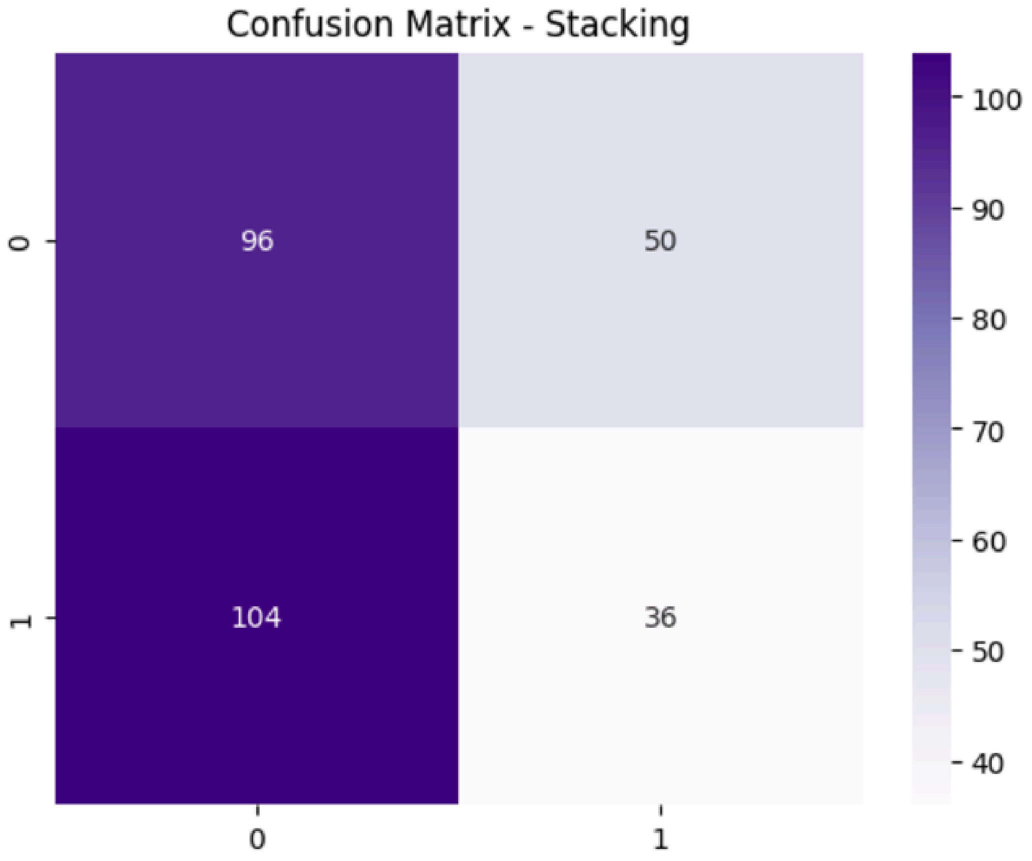
```
LightGBM Classification Report:
              precision    recall  f1-score   support

           0       0.49      0.56      0.52       140
           1       0.51      0.43      0.47       146

    accuracy                           0.50       286
   macro avg       0.50      0.50      0.49       286
weighted avg       0.50      0.50      0.49       286
```



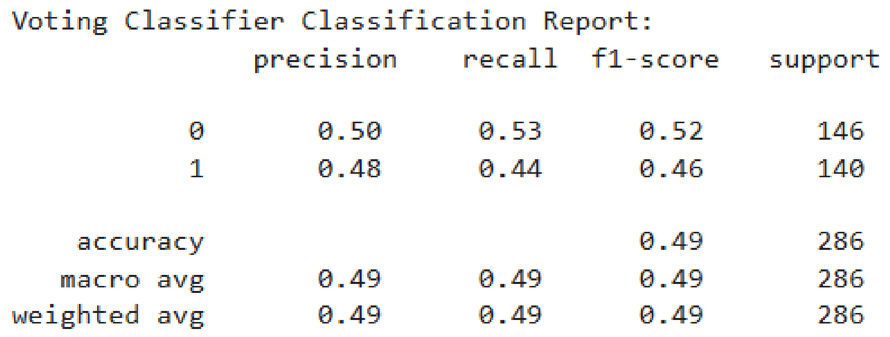
Stacking: Combined strengths of base models and reached an accuracy of 46%.



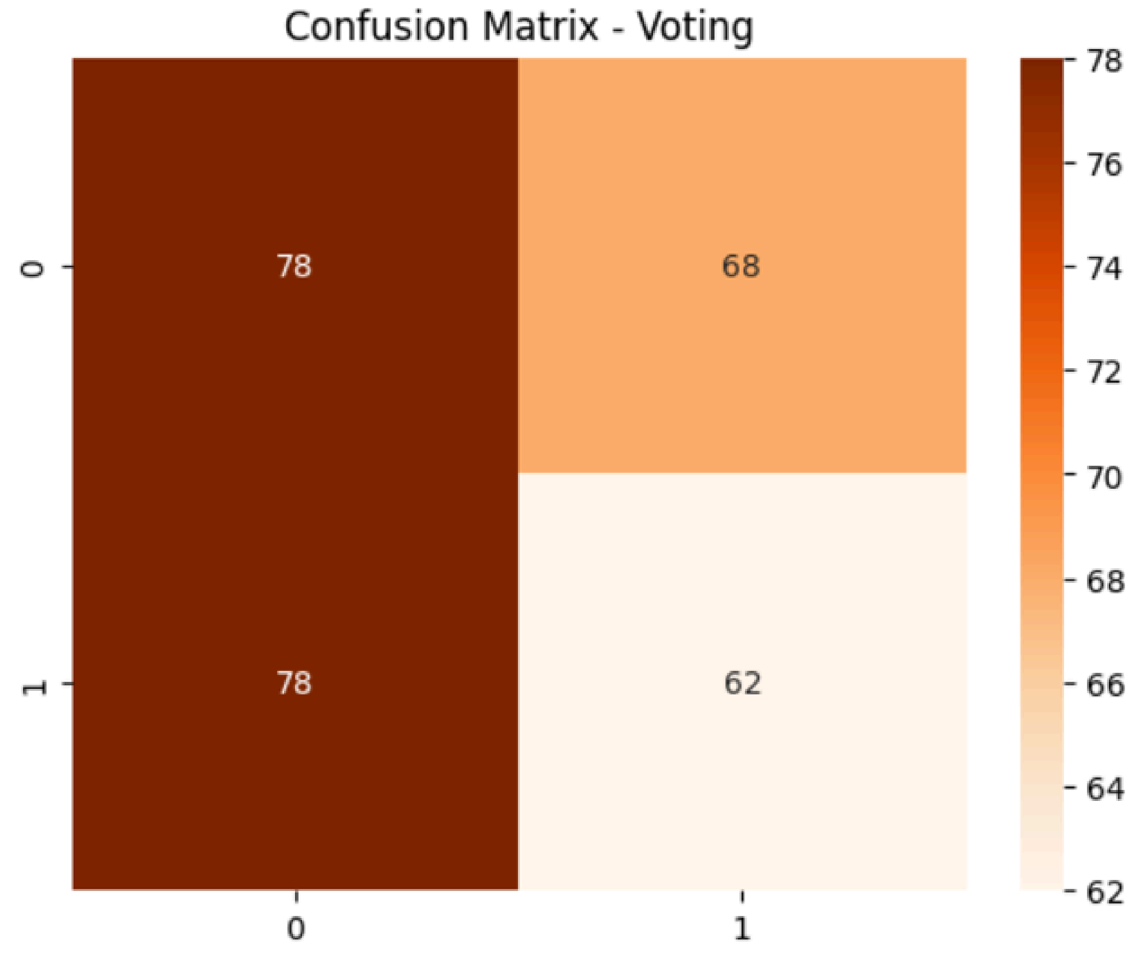
```
Stacking Classification Report:
              precision    recall  f1-score   support

           0       0.48      0.66      0.55       146
           1       0.42      0.26      0.32       140

    accuracy                           0.46       286
   macro avg       0.45      0.46      0.44       286
weighted avg       0.45      0.46      0.44       286
```



Voting: Soft voting provided an accuracy of 49%, balancing precision and recall effectively.

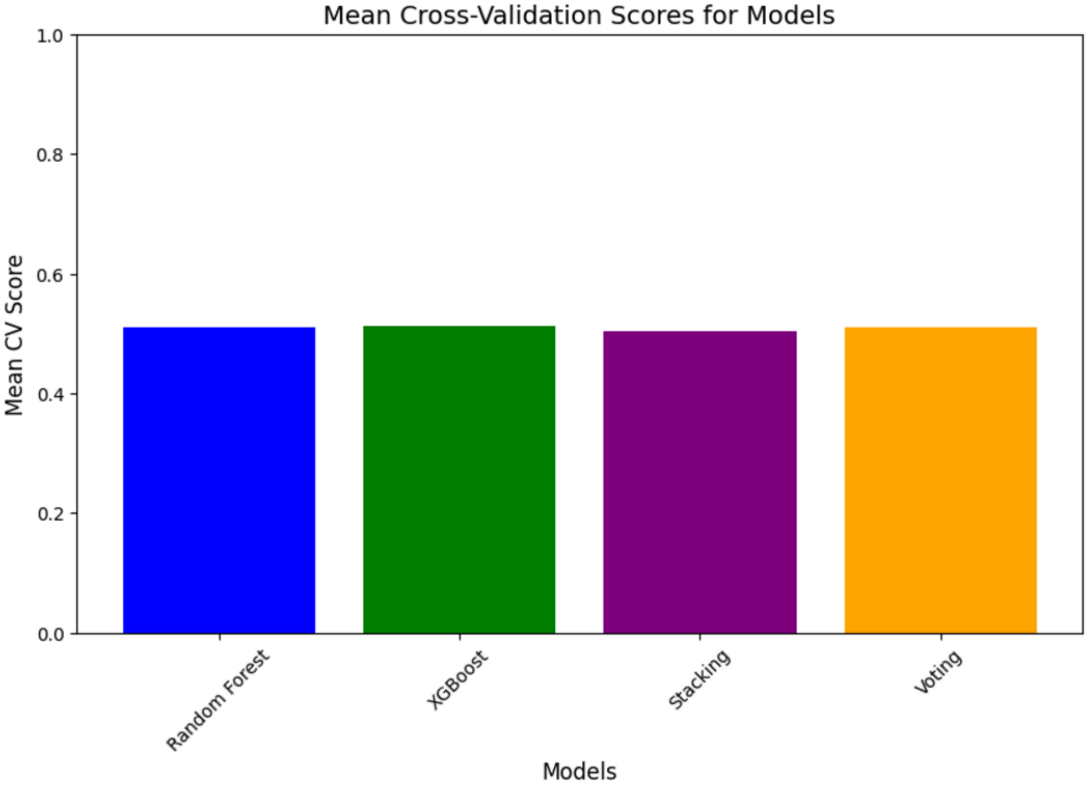


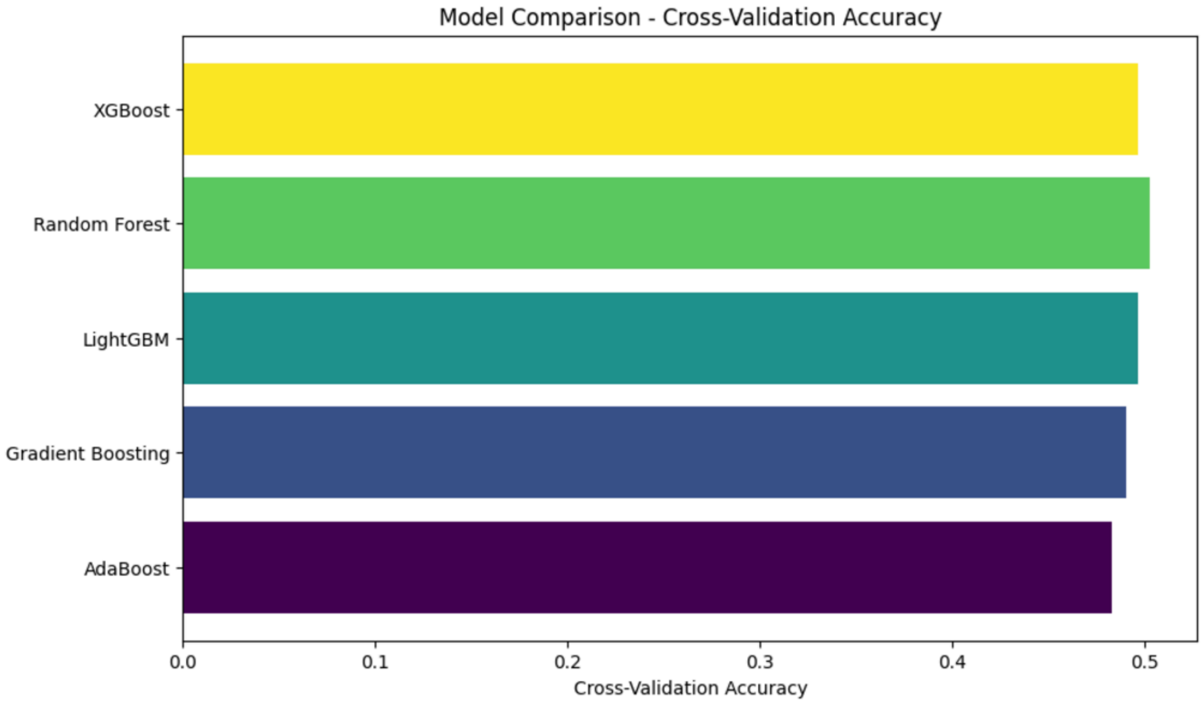
```
Voting Classifier Classification Report:
              precision    recall  f1-score   support

           0       0.50      0.53      0.52       146
           1       0.48      0.44      0.46       140

    accuracy                           0.49       286
   macro avg       0.49      0.49      0.49       286
weighted avg       0.49      0.49      0.49       286
```



2. **Cross validation:** Cross-validation is a resampling method that assesses the performance of a model on an unseen dataset by splitting the entire dataset into multiple subsets-folds-and training and testing the model on different

combinations of these folds. It ensures that the performance of the model generalizes and is not overfitted to a particular split of the dataset.

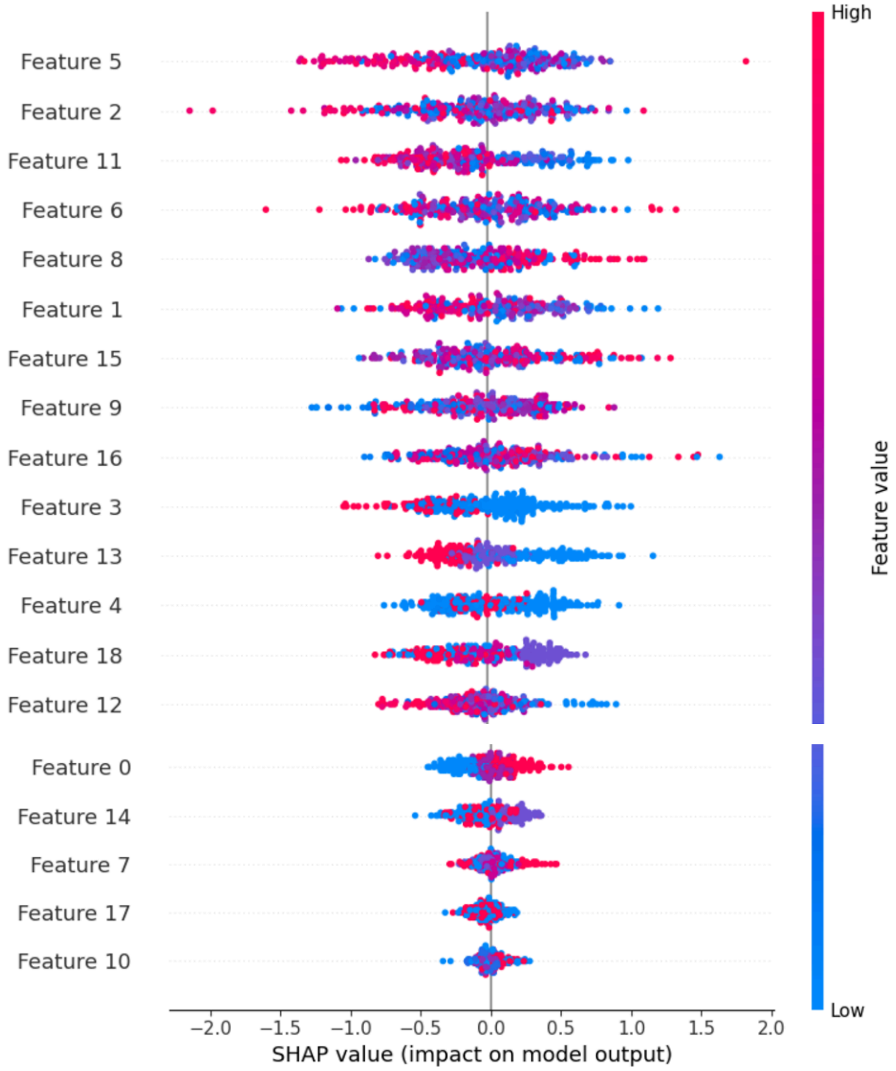




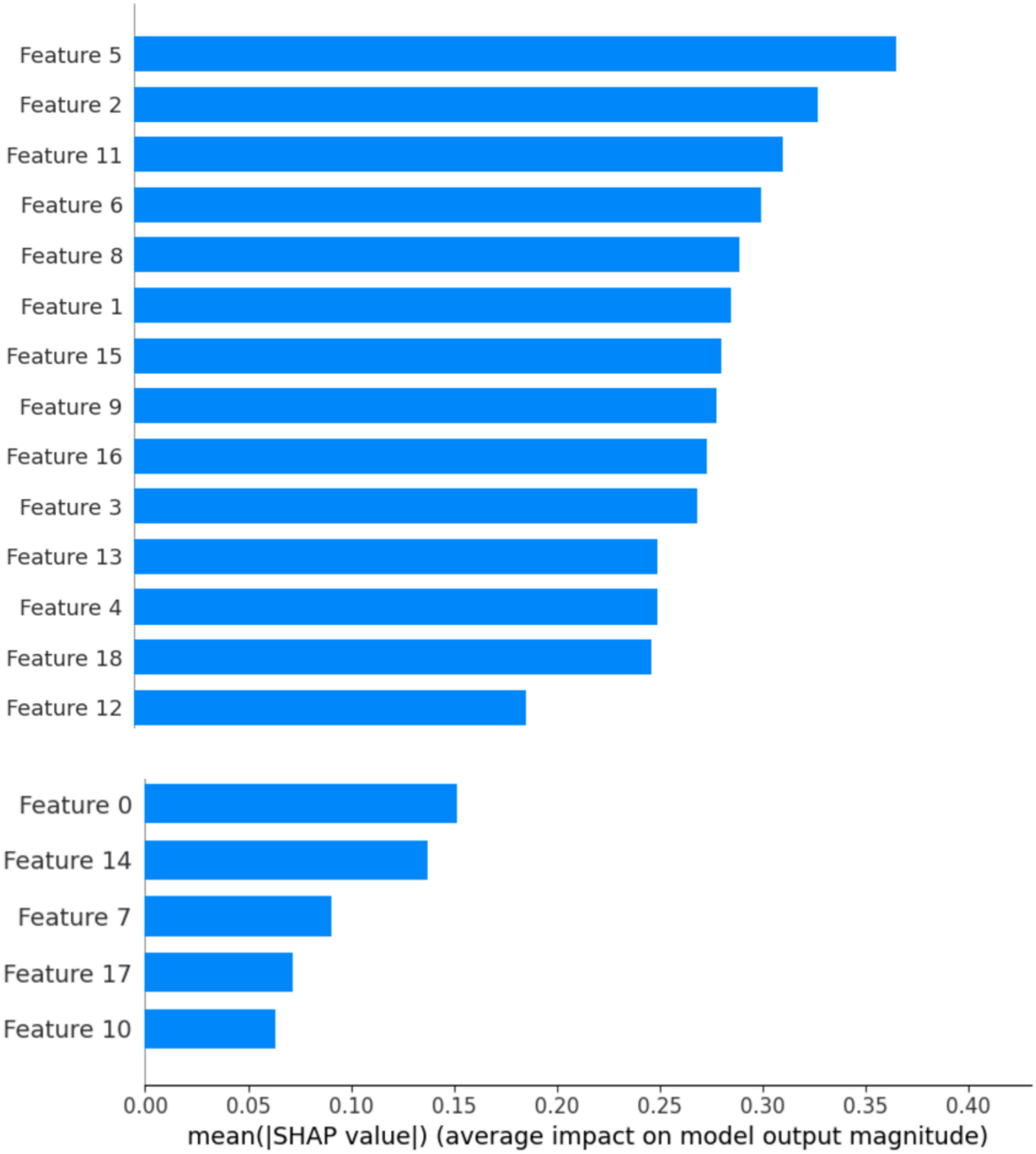
3. **Explainability**:

SHAP and LIME visualizations highlighted features like Bytes_Sent, Protocol, and Flags as significant contributors to predictions.

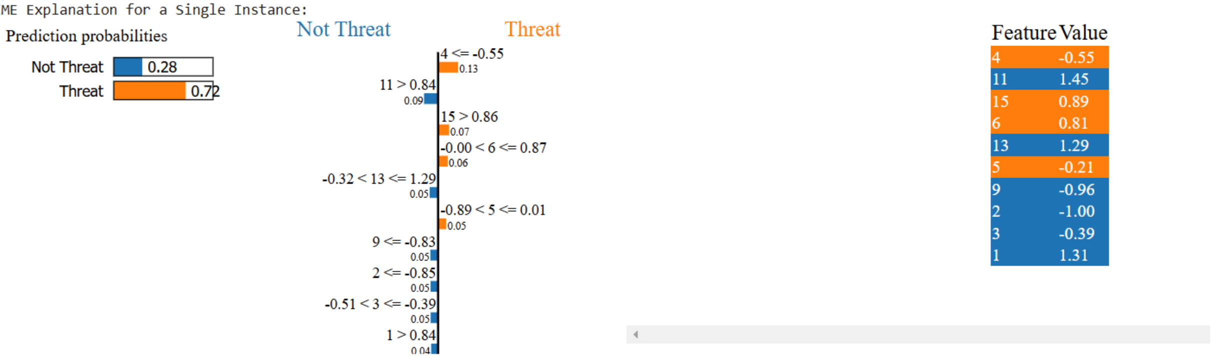
```
Explainable AI with SHAP
SHAP Summary Plot:
```



```
SHAP Feature Importance Plot:
```



```
Explainable AI with LIME
LIME Explanation for a Single Instance:
```

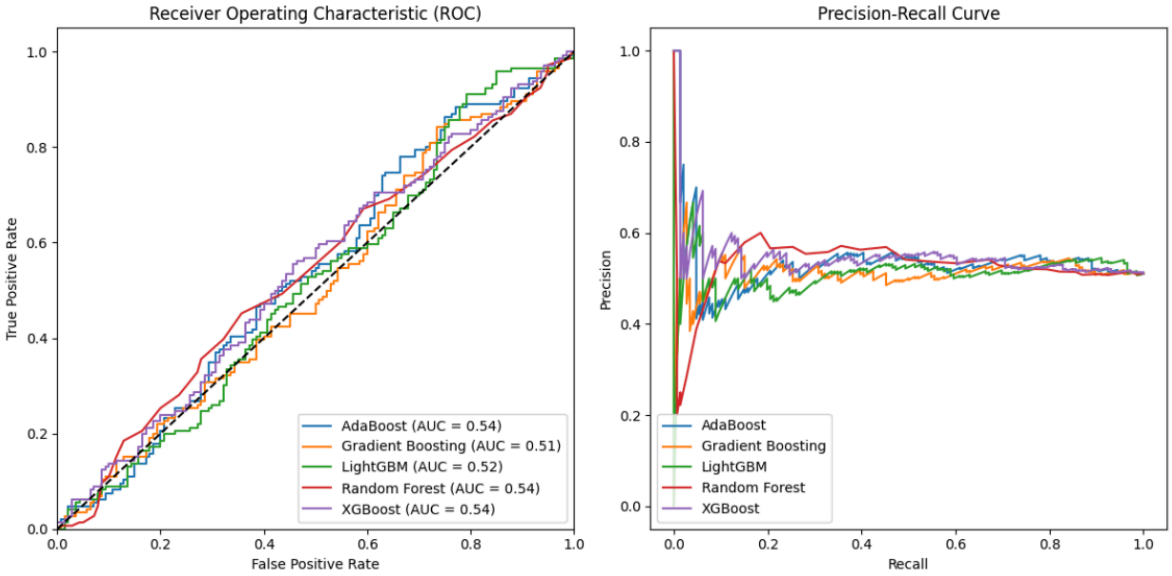


Feature	Value
4	-0.55
11	1.45
15	0.89
6	0.81
13	1.29
5	-0.21
9	-0.96
2	-1.00
3	-0.39
1	1.31

4. **Curve**

   ROC Curve: Visualizes the trade-off between true positive rate and false positive rate at various thresholds.

   PRC (Precision-Recall Curve): Highlights the balance between precision and recall, especially useful for imbalanced datasets.



Conclusion

The proposed project presents ensemble learning for network-based cyber threat detection, including highly accurate models like XGBoost and Random Forest, and methods such as SHAP to ensure interpretability. Reliability and actionability in the predictions were thus derived by using data preprocessing, feature engineering, and advanced model architectures together. These results highlight the key importance of ensemble learning when considering cybersecurity applications that depend critically on the accuracy and explainability of predictive models. Future work could be on deep learning methods and real-time detection systems for higher impact.