Gradient Boosting for IDS

Dataset used: : Intrusion_data.csv

Dataset Split: 80% training and 20% testing

Introduction:

In this project, using the provided dataset, advanced boosting techniques such as Gradient Boosting for IDS have been implemented to classify data as either normal or abnormal. In addition, it follows a structured machine learning process and involves various steps, from data preparation and model training to cross-validation, hyperparameter tuning, and performance evaluation. Specifically, hyperparameters such as, learning rate, maximum depth (max_depth), and number of trees (n_estimators) were implemented in order to improve the model's accuracy and reduce overfitting. Furthermore, metrics such as ROC-AUC and confusion matrix were used to evaluate the overall model's performance.

Data Preparation:

This involves all the data preprocessing steps that are fundamental for preparing mixed data for machine learning. This process basically ensures that the model can interpret and learn from both numerical and categorical information effectively.

Moreover, in the data prepossessing steps, the Gradient Boosting Classifier was integrated into a Pipeline to streamline the machine learning workflow. This Pipeline first applies preprocessing steps to the data, followed by the classifiers.

Numerical Feature Handling:

In this process, StandardScaler was used in order to scale numerical features from the dataset. This step is very crucial for the model's performance as it can help in optimization and

regularization by bringing all the features to a similar scale. This process helps prevent features with larger values from dominating the learning process.

Categorical Feature Handling:

In contrast to the numerical feature handling, this step used OneHoteEncoder to convert categorical into numerical format, so that the machine learning algorithm can process. It involves creation of binary columns for each category.

Furthermore, handle_unknown='ignore' was used to ensure that new categories encountered during the process would not cause any error.

Data Split

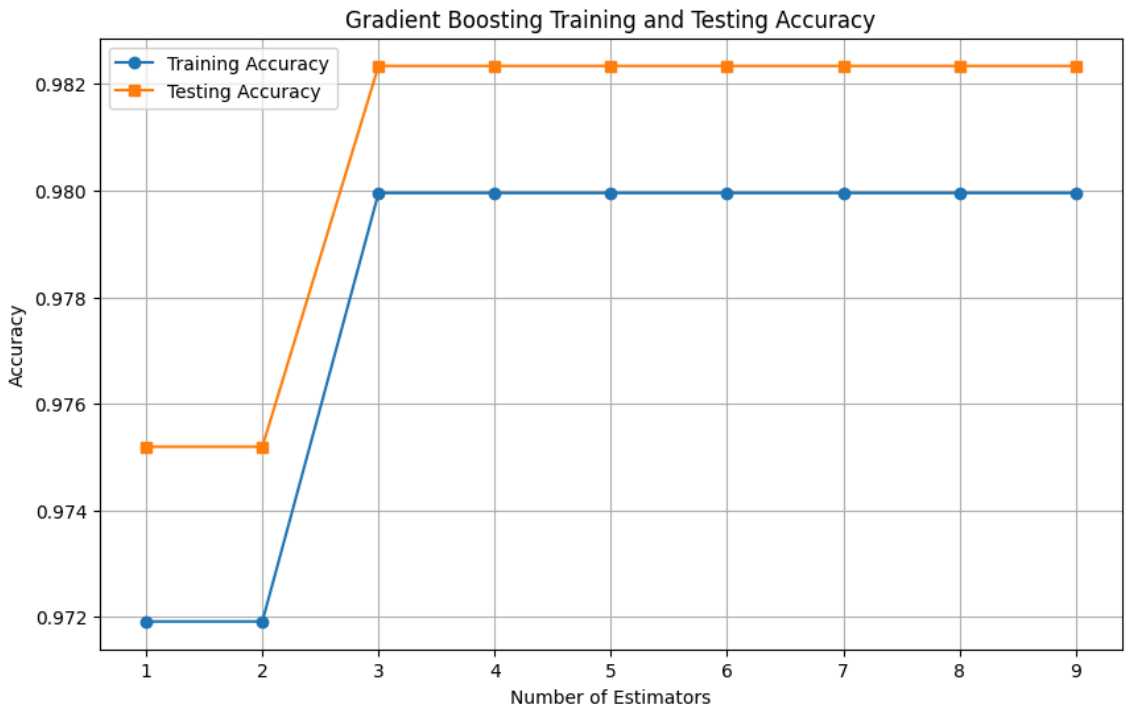
In this case, the dataset was split as:

- 80% training data
- 20% testing data

Model Training and Performance Trends:

The Gradient Boosting model was initially trained by iteratively increasing the number of estimators (trees) from 1 to 9. The plot of training and testing accuracies against the number of estimators showed a consistent trend:

- Both training and testing accuracies increased as the number of estimators increased
- The accuracies of both training and testing converges rapidly, which indicates that even a small number of estimators were effective in determining the patterns in the data.
- Overall, there was no significant gap between training and testing accuracy.



Cross-Validation Results:

To ensure the model's generalization capability and effectiveness, a 5-fold cross-validation was performed on the entire dataset. The results were:

- **Cross-Validation Scores:** [0.97856718, 0.98253622, 0.98034934, 0.97995236, 0.98034934]
- **Average Cross-Validation Score:** 0.9803508891070454

This results suggests that the Gradient Boosting model is performing very well and generalizes consistently across different subsets of your data, indicating high accuracy in detecting intrusions.

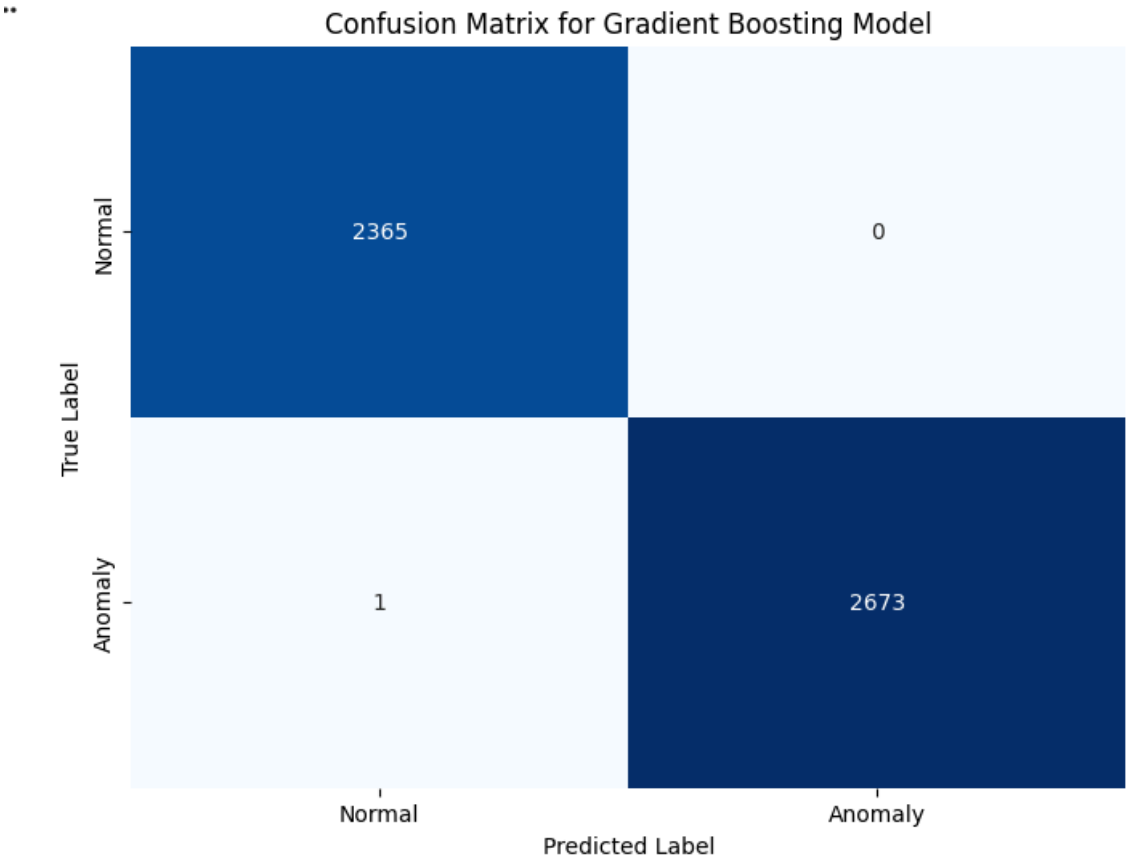
Hyperparameter Tuning

The hyperparameter tuning for the Gradient Boosting model was performed using GridSearchCV with 5-fold cross-validation, utilizing the ROC-AUC score as the evaluation metric. For simplicity, the tuning process focused on three key hyperparameters: learning rate, maximum depth (max_depth), and the number of estimators (n_estimators).

- **Best Parameters Found:** Learning_rate: 0.1, max_depth: 5, n_estimator': 100}
- **Best Cross-Validation ROC-AUC Score:** 0.9997343129250222

The ROC-AUC score of 0.9997 suggests that the model can reliably separate positive and negative classes with high accuracy.

Confusion Matrix Interpretation:



- High True Positives: 2673
    - The model correctly identified 2673 instances of anomaly or intrusion activities.
- False Positives: 0
    - The model incorrectly classified 0 instances of normal activity.
- False Negatives: 1
    - The model incorrectly classified only 1 instance of intrusion as normal activity.
- High True Negative: 2365
    - The model correctly identified 2365 instances of normal activity.

The confusion matrix demonstrates exceptional performance by this model. This suggests that almost every actual intrusion was successfully detected while non-intrusions were detected as normal, with 0 False positives and 1 False Negatives. This is an example of a very good model for an Intrusion Detection System.

Conclusion:

Overall Finding and Analysis

After data preprocessing and hyperparameter tuning, the Gradient Boosting Classifier has demonstrated outstanding results and performance in detecting network intrusions on the given dataset. With a high average cross validation ROC-AUC score, the model is proven to be highly accurate and reliable for Intrusion Detection Systems.