<u>NETWORK SECURITY PACKAGE</u>
Intrusion Detection System

-Sree Varshni S
22PC35

ABSTRACT:

Intrusion Detection Systems (IDS) are critical for identifying and mitigating malicious activities within networks. This study focuses on building an IDS using the KDD dataset, which is widely used for benchmarking intrusion detection algorithms. The dataset is preprocessed to handle missing values, standardize features, and encode categorical data to ensure optimal model performance.

Four machine learning models—Random Forest, Naive Bayes, Decision Tree, and Gradient Boosting—are employed to classify network traffic as either normal or anomalous. Each model leverages its unique strengths: Random Forest and Gradient Boosting offer robust ensemble methods, while Decision Tree and Naive Bayes provide fast, interpretable results. The trained models are evaluated based on accuracy and detection rates to determine the most effective approach for intrusion detection.

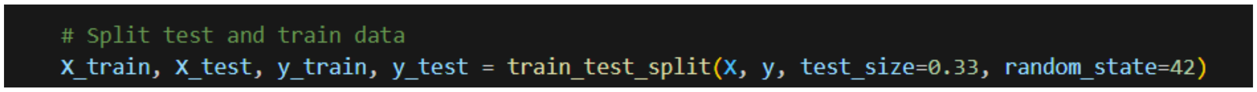
MODEL PREPARATION:

1)Data Preprocessing: This phase focuses on cleaning and preparing the dataset for analysis. It includes removing highly correlated variables to avoid redundancy and improve data quality. The following columns were dropped based on correlation analysis:

- **num_root** (correlated with **num_compromised**: 0.9938)
- **srv_serror_rate** (correlated with **serror_rate**: 0.9984)
- **srv_rerror_rate** (correlated with **rerror_rate**: 0.9947)
- **dst_host_srv_serror_rate** (correlated with **srv_serror_rate**: 0.9993)
- **dst_host_serror_rate** (correlated with **rerror_rate**: 0.9870)
- **dst_host_rerror_rate** (correlated with **srv_rerror_rate**: 0.9822)
- **dst_host_srv_rerror_rate** (correlated with **rerror_rate**: 0.9852)
- **dst_host_same_srv_rate** (correlated with **srv_rerror_rate**: 0.9866)
- **is_guest_login**
- **srv_count**
- **dst_host_srv_count**

2)Feature Selection: After preprocessing, feature selection identifies the most important features that contribute to prediction. While the removal of certain columns due to high correlation is part of this process, the focus is on retaining the most relevant features for model training

3)**Data Splitting**: The dataset was split into training and testing sets to evaluate the model's performance. The training set comprises 67% of the data, while the testing set comprises 33%. The split is executed as follows:
python.



```
# Split test and train data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33, random_state=42)
```

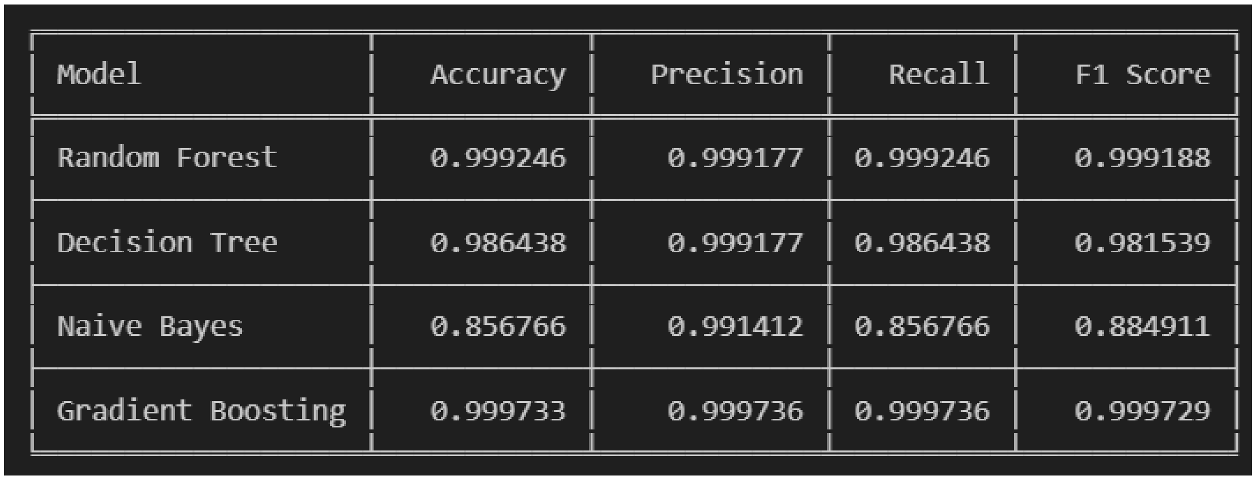
Overall, these steps enhance the dataset's quality and relevance, setting the stage for effective model training in the intrusion detection system.

MODELS USED

1) **Naive Bayes**: This model is based on applying Bayes' theorem with the assumption of independence among predictors.
    - **Precision**: 0.991412
    - **Accuracy**: 0.856766
    - **F1 Score**: 0.884911
    - **Recall**: 0.856766
2) **Decision Tree**: A non-parametric model that splits the data into branches based on feature values, making decisions based on a series of questions.
    - **Precision**: 0.999177
    - **Accuracy**: 0.986438
    - **F1 Score**: 0.981539
    - **Recall**: 0.986438
3) **Random Forest**: An ensemble method that builds multiple decision trees and merges them to improve accuracy and control overfitting.
    - **Precision**: 0.999177
    - **Accuracy**: 0.999246
    - **F1 Score**: 0.999188
    - **Recall**: 0.999246
4) **Gradient Boosting**: An ensemble technique that builds trees sequentially, with each tree correcting errors made by the previous ones, often leading to high accuracy.
    - **Precision**: 0.999736
    - **Accuracy**: 0.999733

- **F1 Score**: 0.999729
- **Recall**: 0.999736

Model performance:

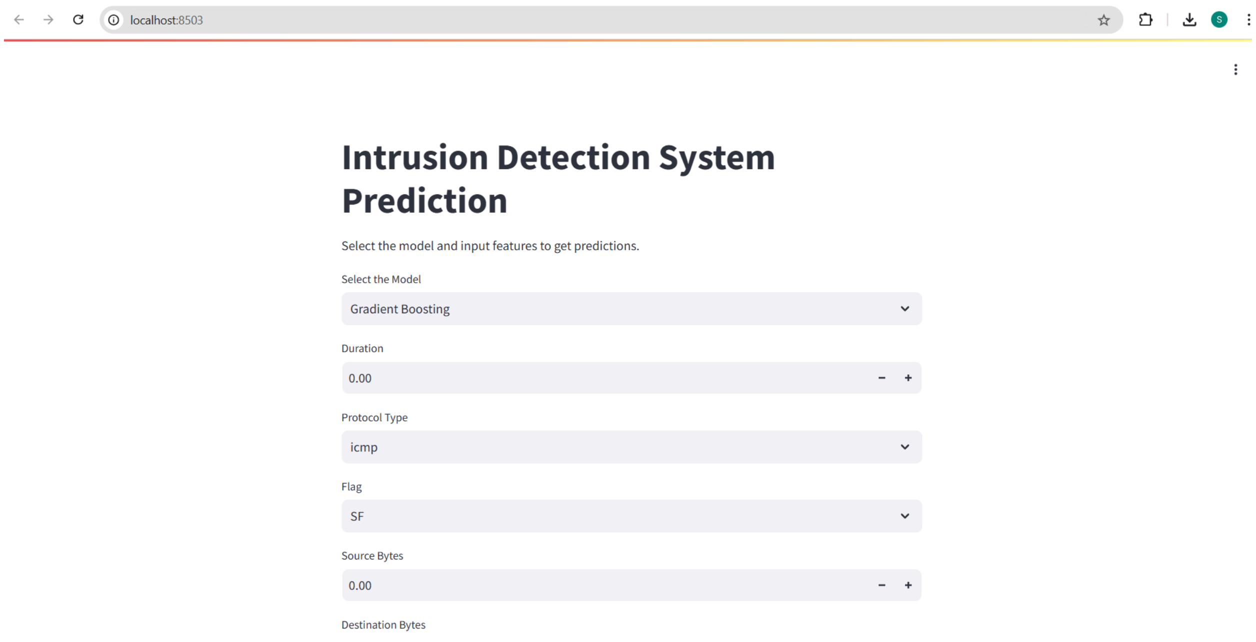


```
Model              Accuracy   Precision   Recall     F1 Score
Random Forest      0.999246   0.999177    0.999246   0.999188
Decision Tree      0.986438   0.999177    0.986438   0.981539
Naive Bayes        0.856766   0.991412    0.856766   0.884911
Gradient Boosting  0.999733   0.999736    0.999736   0.999729
```

Streamlit Deployment:

This project leverages Streamlit to build an interactive web application for the Intrusion Detection System (IDS). Users can input features from the KDD dataset—like duration, protocol type, and counts—and select from trained models (Random Forest, Gradient Boosting, Decision Tree, Naive Bayes) for predictions.

Upon clicking "Predict," users receive results, including the predicted attack type and category, along with error handling for any issues. This deployment simplifies user interaction with IDS models and enables real-time prediction visualization.



localhost:8503

Intrusion Detection System Prediction

Select the model and input features to get predictions.

Select the Model

Gradient Boosting

Duration

0.00

Protocol Type

icmp

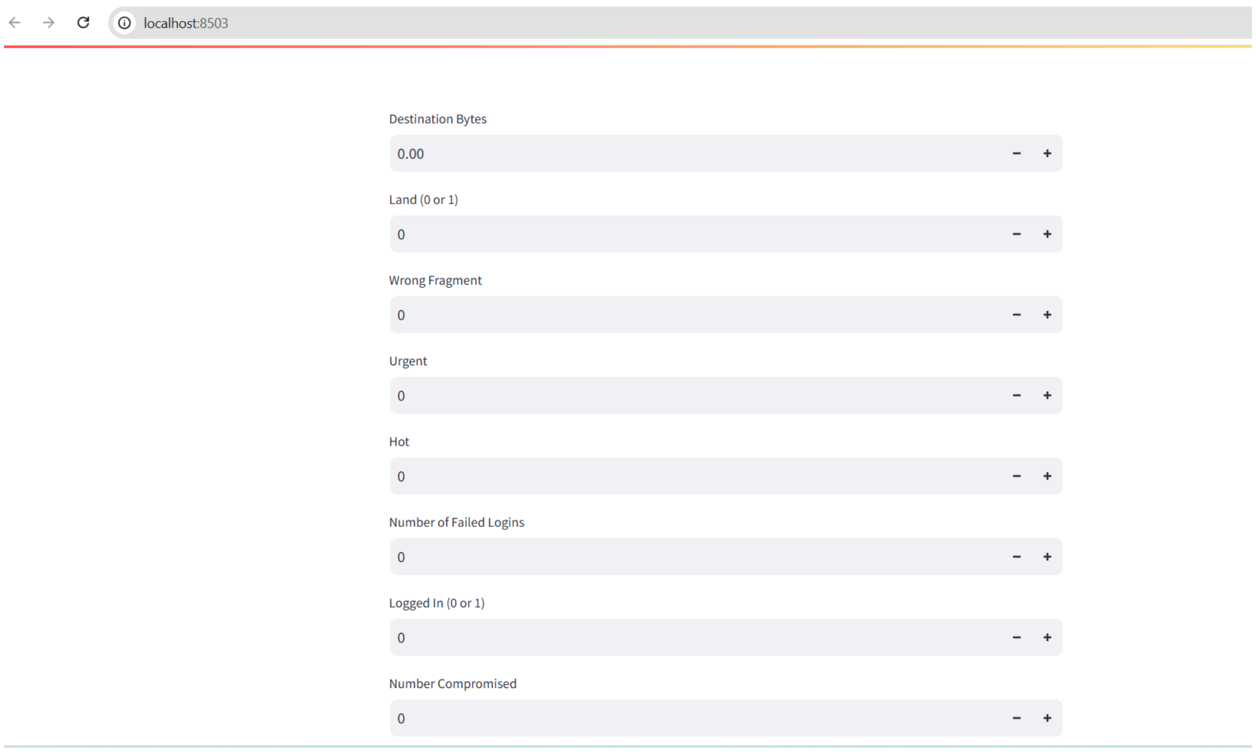
Flag

SF

Source Bytes

0.00

Destination Bytes



localhost:8503

Destination Bytes

0.00

Land (0 or 1)

0

Wrong Fragment

0

Urgent

0

Hot

0

Number of Failed Logins

0

Logged In (0 or 1)

0

Number Compromised

0

Root Shell (0 or 1)

0

SU Attempted (0 or 1)

0

Number of File Creations

0

Number of Shells

0

Number of Access Files

0

Number of Outbound Commands

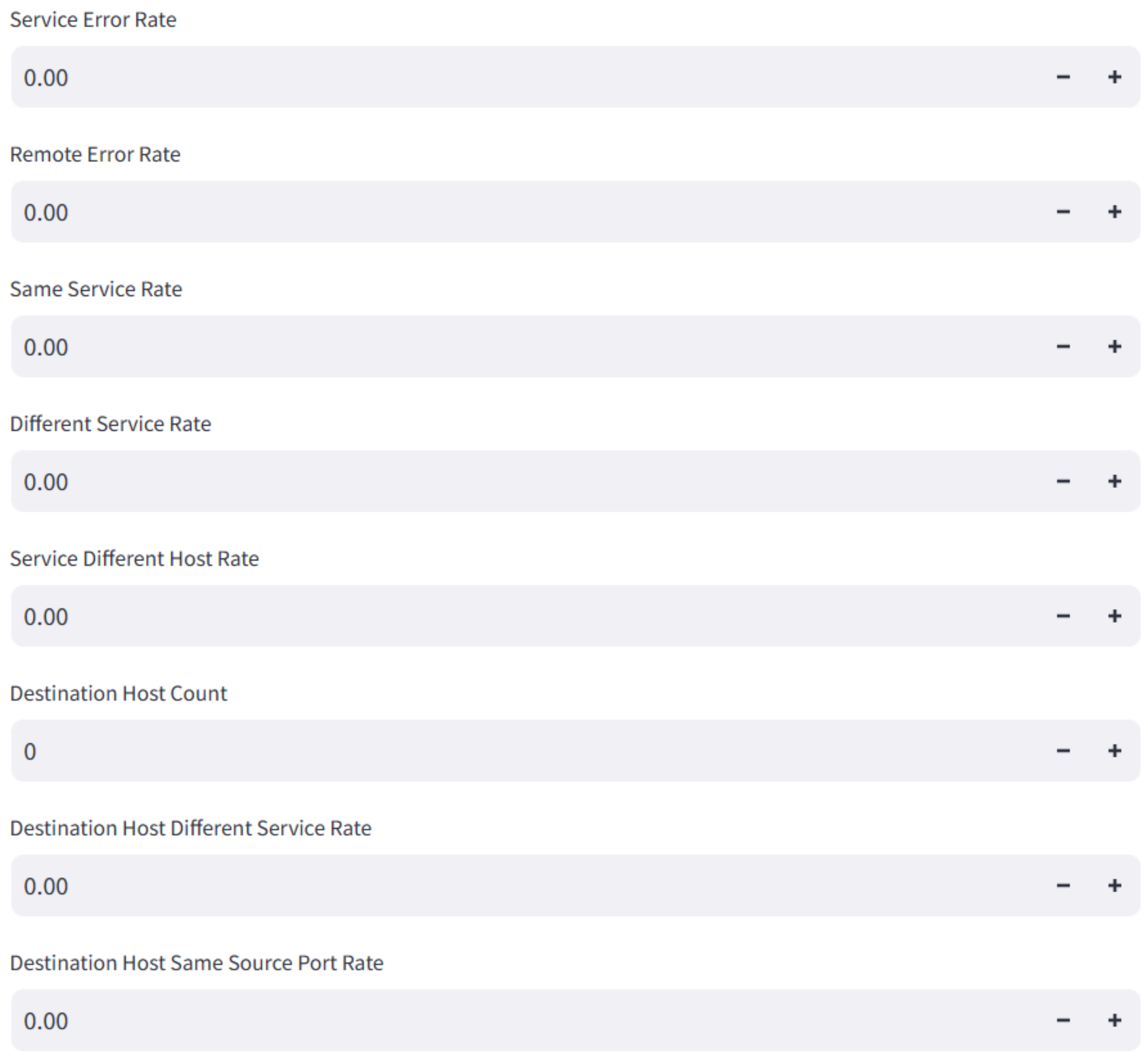
0

Is Host Login (0 or 1)

0

Count

0


Service Error Rate

0.00

Remote Error Rate

0.00

Same Service Rate

0.00

Different Service Rate

0.00

Service Different Host Rate

0.00

Destination Host Count

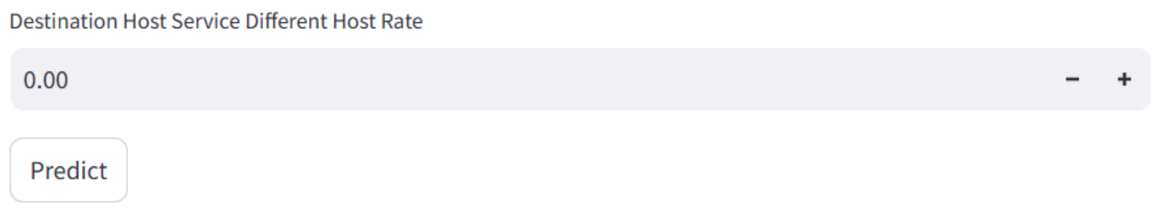
0

Destination Host Different Service Rate

0.00

Destination Host Same Source Port Rate

0.00


Destination Host Service Different Host Rate

0.00

CONCLUSION:

In conclusion, this Intrusion Detection System project effectively demonstrates the application of machine learning models to identify potential security threats. By preprocessing the KDD dataset and implementing various models, we achieved high accuracy and performance metrics. The interactive Streamlit application allows users to easily input data and receive predictions, showcasing a practical approach to enhancing cybersecurity measures. Overall, this project highlights the importance of data-driven solutions in protecting networks from intrusions.

GITHUB LINK:

https://github.com/SREEVARSHNI18/Intrusion-Detection-System.git