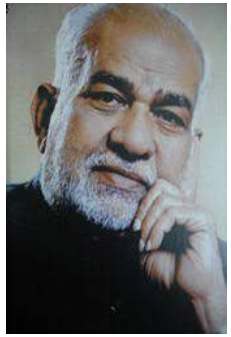
BVRIT Hyderabad
College Of Engineering For Women



KEYLOGGER DETECTION

Team-10

Sk.Ramjanbee-20WH1A05H2

V.Poojitha- 20WH1A05H3

J.Renatus Beatrice-20WH1A05H5

M.Bhavana- 20WH1A05H6

T.Sriya-20WH1A05H7

April 1, 2023

Problem Statement

Detection of Keylogger using Machine Learning

- perform data exploration, preprocessing and visualization
- implement classification model using sklearn library
- evaluate the model using appropriate performance metrics
- develop the Keylogger Prediction system

Python Packages and Libraries used

- Pandas
- Matplotlib
- seaborn
- sklearn
- Numpy
- tkinter

Algorithms

- Decision Tree
- K Nearest Neighbor
- Random Forest

Decision Tree

Decision tree is a fundamental component of the random forest algorithm. Decision tree works by recursively splitting the data into subsets based on the most significant feature, which results in a tree-like structure. Each internal node of the tree represents a test on a feature, and each branch represents the outcome of the test. The leaves of the tree represent the class labels. In the context of keylogger detection, decision tree can be used to classify different types of attacks based on their features, such as the behavior of the user and the system.

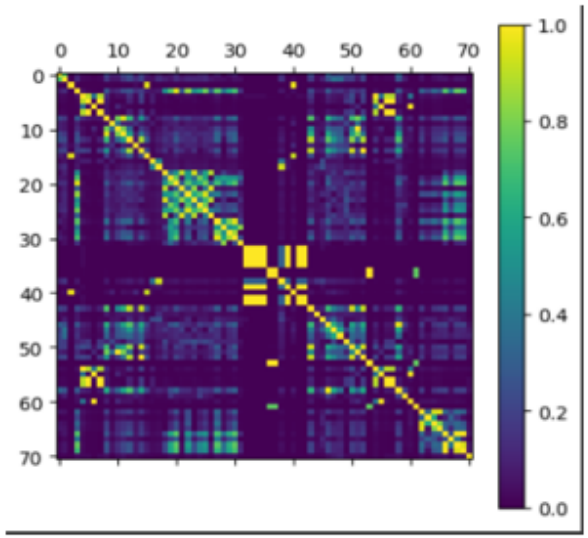
K Nearest Neighbor

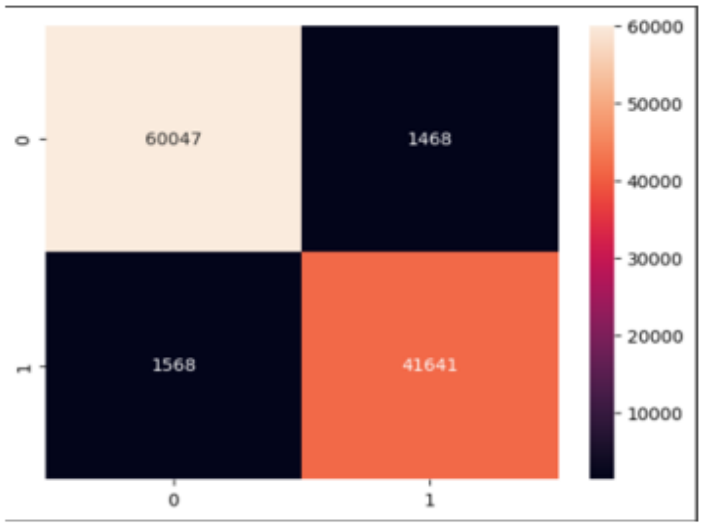
KNN (K-Nearest Neighbors) is a machine learning algorithm that can be used for keylogger detection. KNN works by finding the k-nearest neighbors to a given data point and classifying it based on the majority class of those neighbors.By analyzing the behavior of the keylogger, KNN can identify different keystrokes and patterns generated by the keylogger. KNN may not be the most suitable algorithm for keylogger detection in some cases.

Random Forest

In keylogger detection, the random forest algorithm works by combining the output of multiple decision trees to classify different types of attacks and improve the accuracy of the classifier. Each individual tree in the random forest spits out a class prediction, and the class with the most votes becomes the model's prediction.By using this algorithm, keylogger detection systems can accurately classify different types of attacks and improve their overall accuracy.

Confusion Matrix and Correlation Matrix





Comparison Table

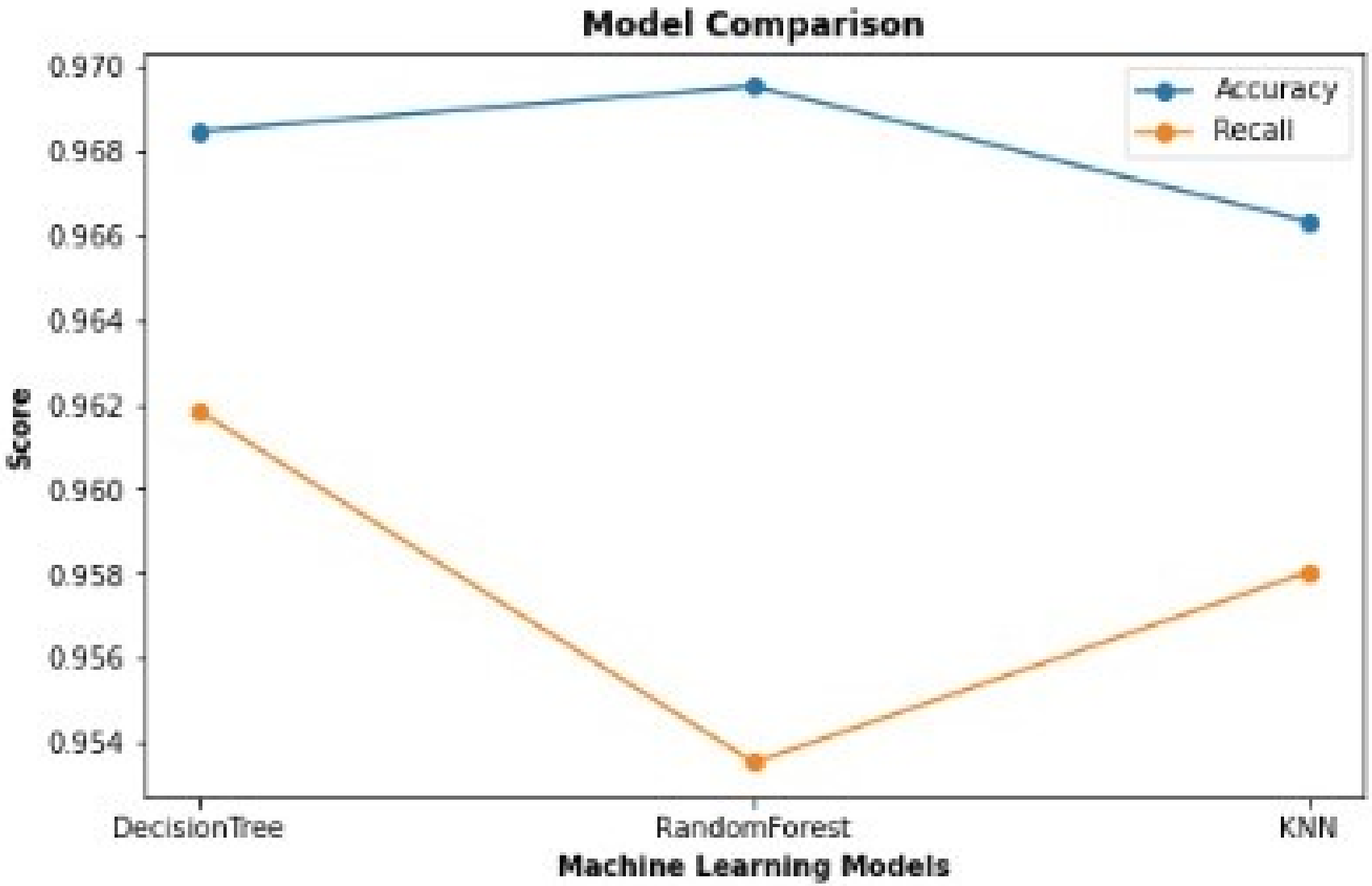
Comparision of different models based on accuracy and recall

```
# Define the models and their accuracy and recall scores
models = ['DecisionTree', 'RandomForest', 'KNN']
accuracy_scores = [dtcaccuracy, rfcaccuracy, accuracy]
recall_scores = [dtcrecall, rfcrecall, recall]
```

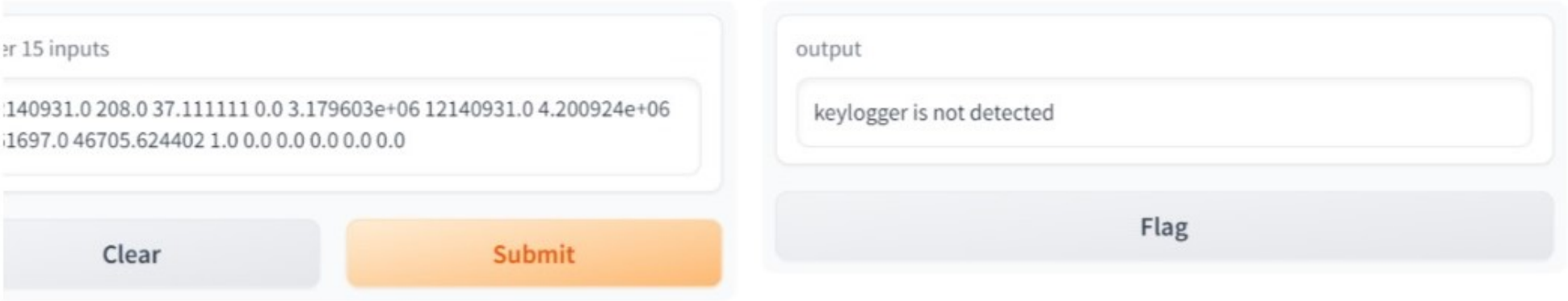
```
[128] comp_tb = pd.DataFrame({'Model': models, 'Accuracy': accuracy_scores, 'Recall': recall_scores})
      print(comp_tb)
```

```
           Model  Accuracy    Recall
0   DecisionTree  0.968450  0.961841
1   RandomForest  0.969512  0.953532
2            KNN  0.966319  0.958042
```

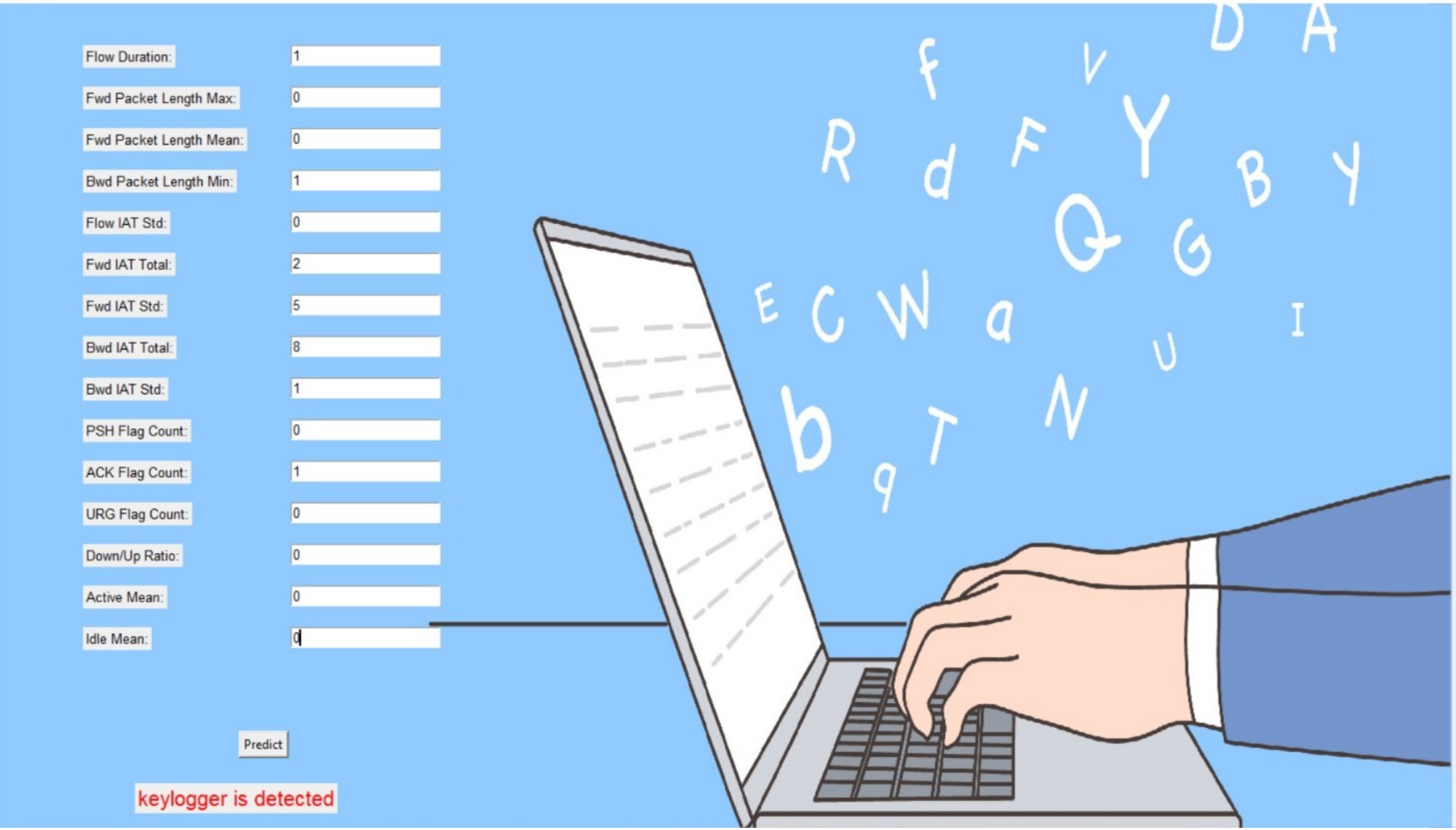
Comparison Graph



Output using Gradio



Output using GUI



Execute code

- COLAB LINK FOR THE CODE

THANK YOU!