Hybrid Machine Learning Model for Efficient Botnet Attack Detection in IoT Environment

Machine learning algorithms performance often enhance by combining power of multiple algorithms and this process known as Hybrid ML. IoT network consists of small devices which sense data from environment and send to centralized server for monitoring using internet network connection. Due to network connectivity IoT always vulnerable to different attacks. Advance antivirus cannot be install in IoT as they run on battery power with small memory so ML or DL algorithms are the alternate option to secure IoT from attacks. ML algorithms get trained on past dataset and then detect attacks from future IoT network data.

In the past many ML or DL algorithms such as ANN, CNN, RNN and many ML algorithms was introduced to detect IoT botnet attacks. This algorithms will detect attacks based on its implemented logic and don’t have power of multiple algorithms logic so its detection accuracy will be less.

To further enhance accuracy author of this paper employing combination or stacking of different algorithms such as ANN + CNN + LSTM + RNN to form a hybrid algorithm and this algorithm accuracy is higher than single traditional algorithms.

To train and test each algorithms performance author has given many hyper-parameters and then trained and test with different epochs and KFOLD but training and testing with all this KFOLD will take lots of time so we trained all algorithms with 32 batch size and number of epoch as 30 and KFOLD value as 1.

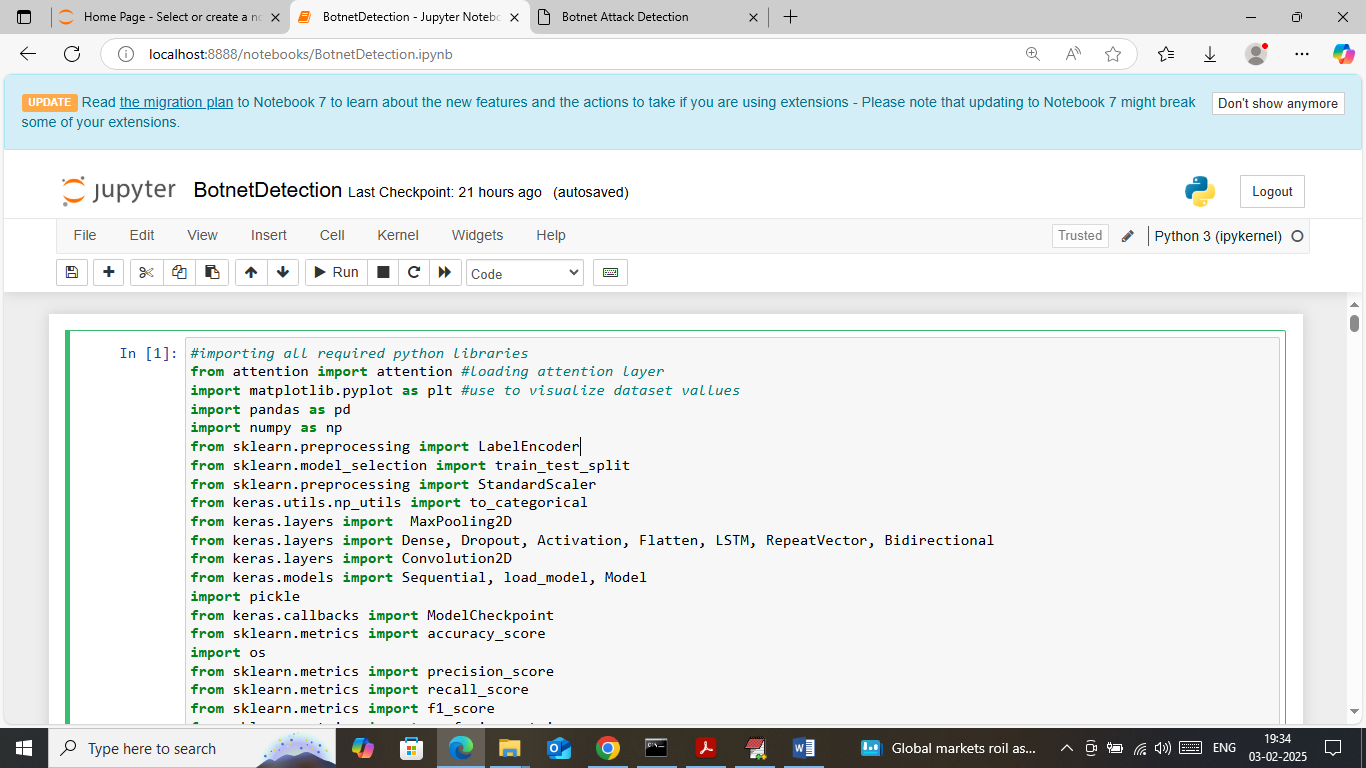
To evaluate each algorithm performance author has used UNSW15 dataset which contains 15 different attacks. Each algorithm get trained on above dataset and calculate different prediction metrics such as Accuracy, Precision, Recall, Confusion matrix, ROC graph and FSCORE. Among all algorithms propose stacking algorithm called ACLR is giving high accuracy.

Extension Concept

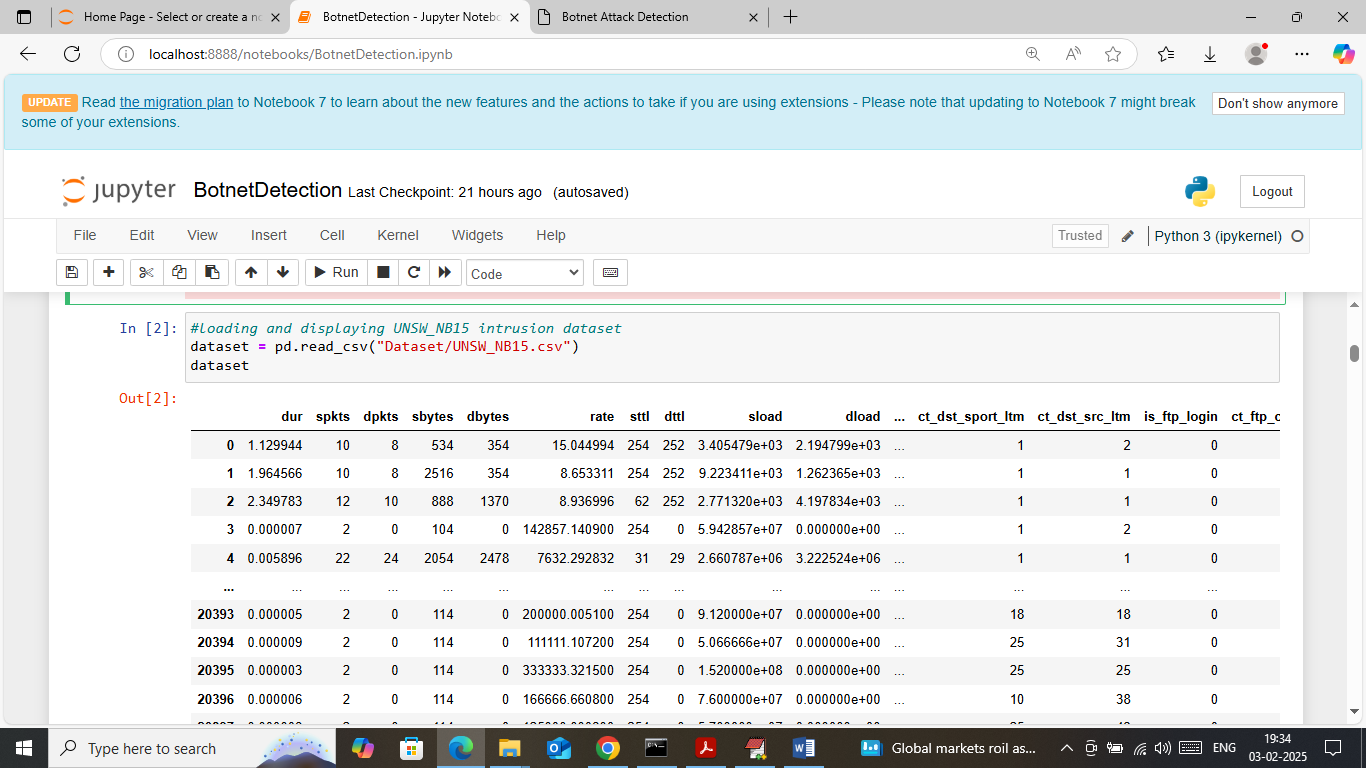
To further enhance detection accuracy we have modified propose ACLR algorithm with ATTENTION mechanism which helps models focus on relevant information in data. It allows models to assign different weights to different parts of the data, so they can pay more attention to the most important parts and can help in better prediction accuracy.

SCREEN SHOTS

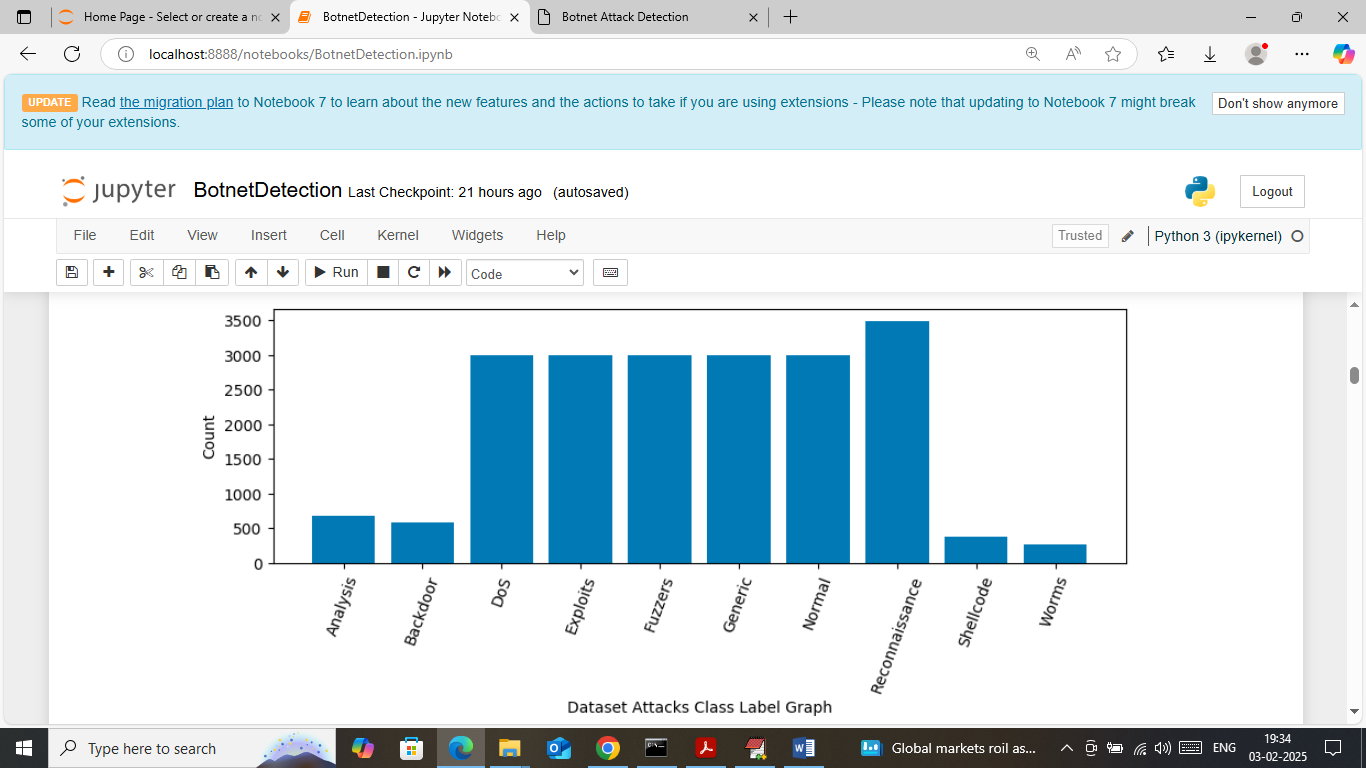
For dataset evaluation, processing, training and testing we have used JUPYTER notebook and to predict attacks from test data we have used FLASK web framework. To start JUPYTER double click on ‘runJUPYTER.bat’ file to get below page. In below screen we can see code and output with blue colour comments



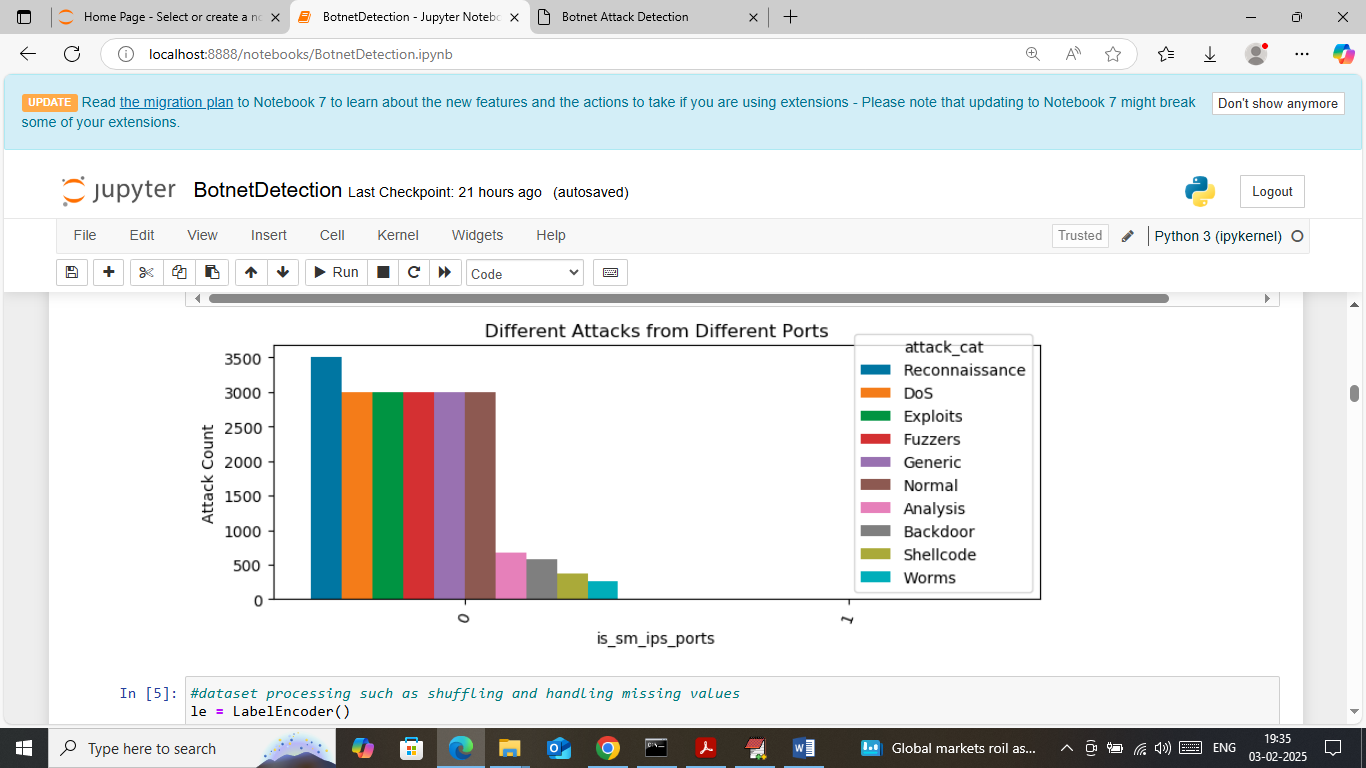
In above screen loading required python classes and packages



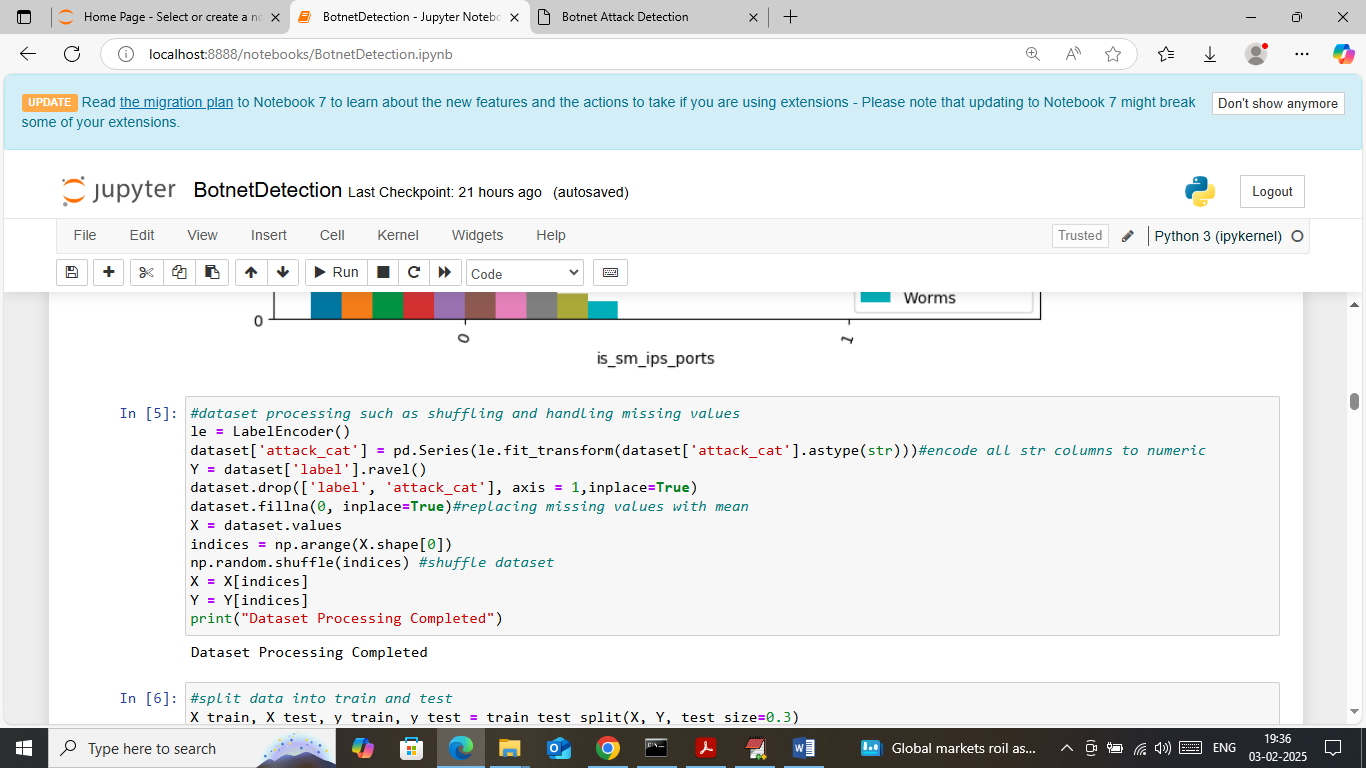
In above screen loading and displaying UNSW IOT dataset



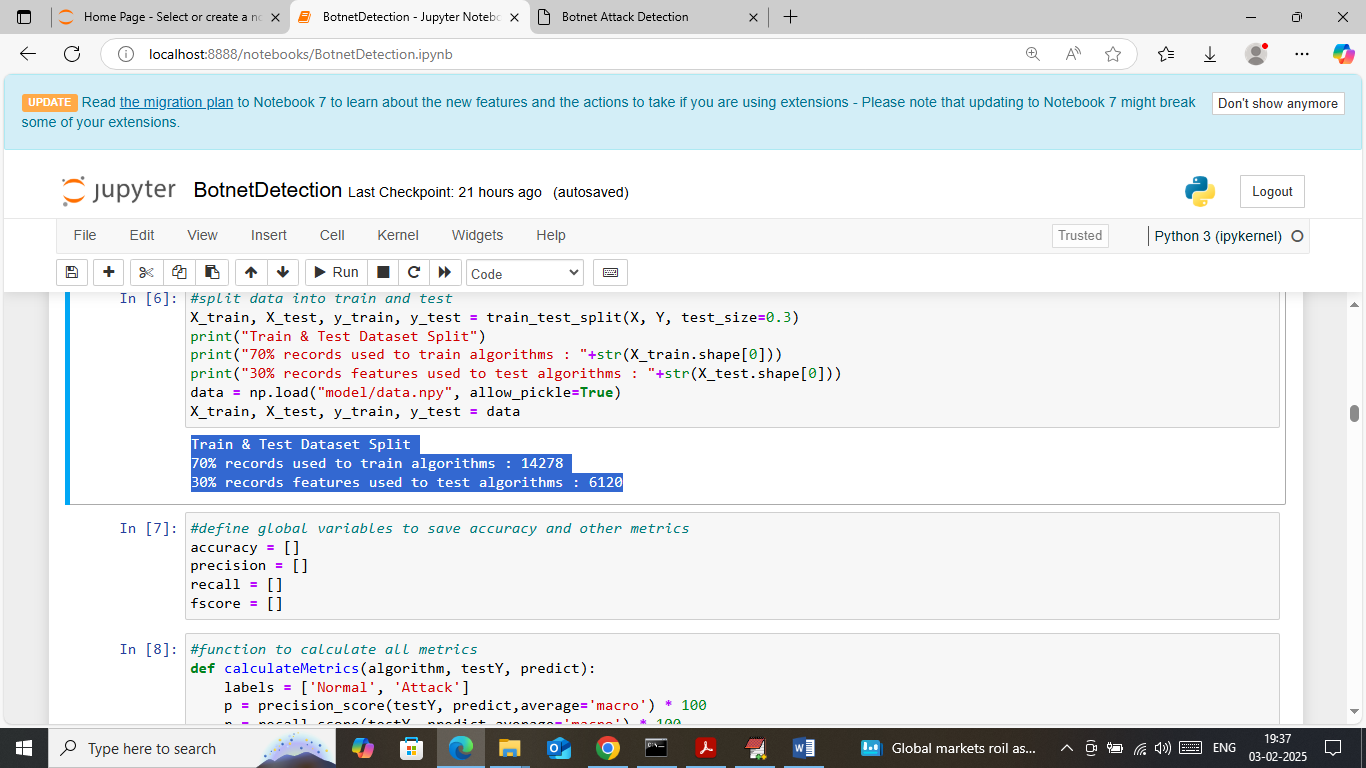
In above screen visualizing graph of different attacks found in dataset where x-axis represents attack name and y-axis represents number of instances



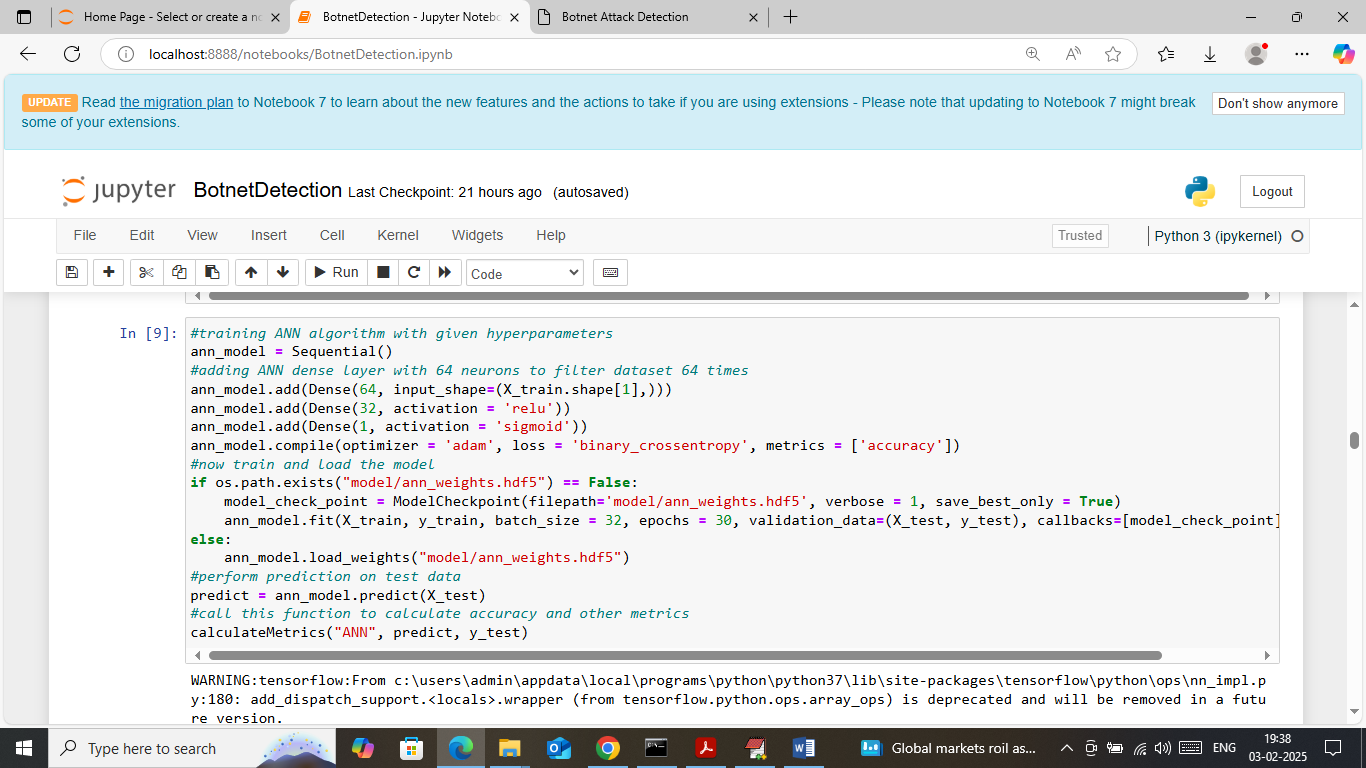
In above screen visualizing graph of different attacks which happen from different ports where x-axis represents PORT No and y-axis represents different attack counts from that port



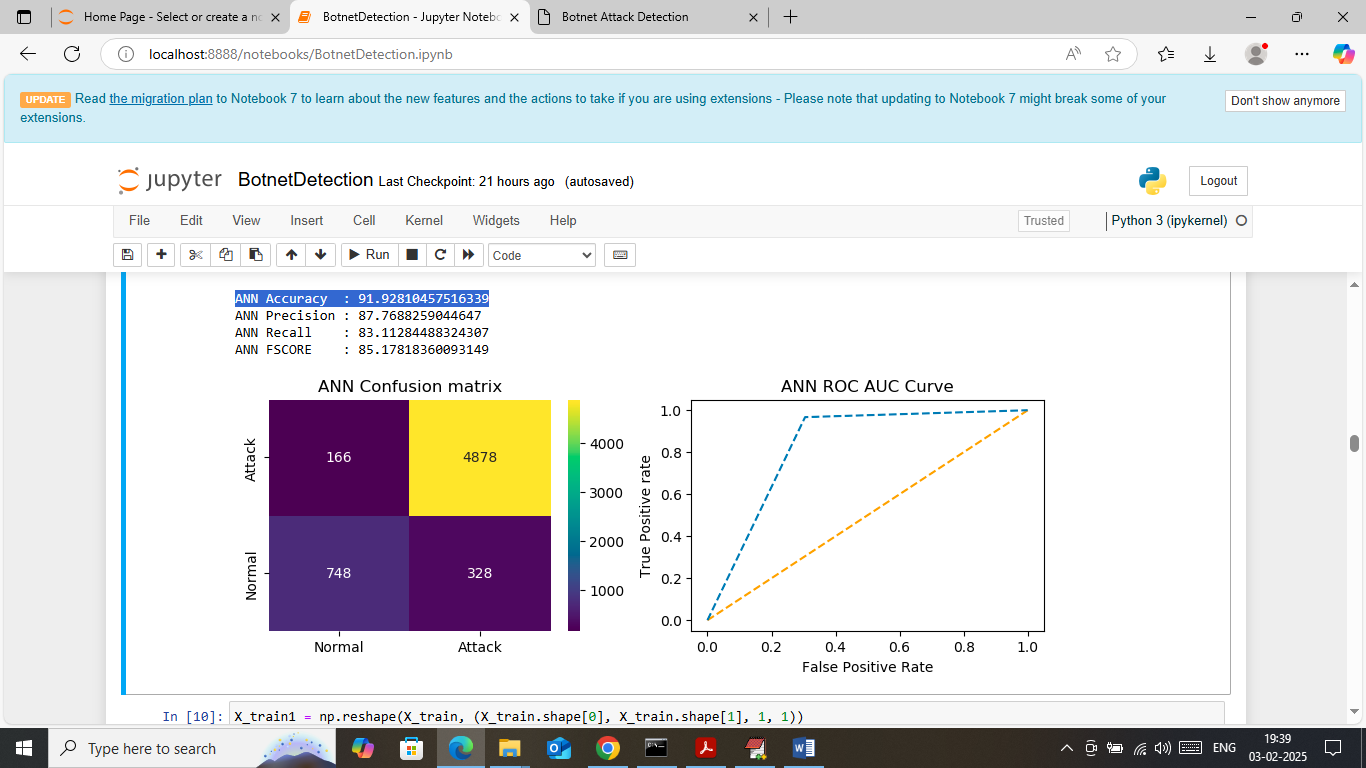
In above screen applying dataset processing techniques such as handling missing values, shuffling and then converting non-numeric label to numeric label



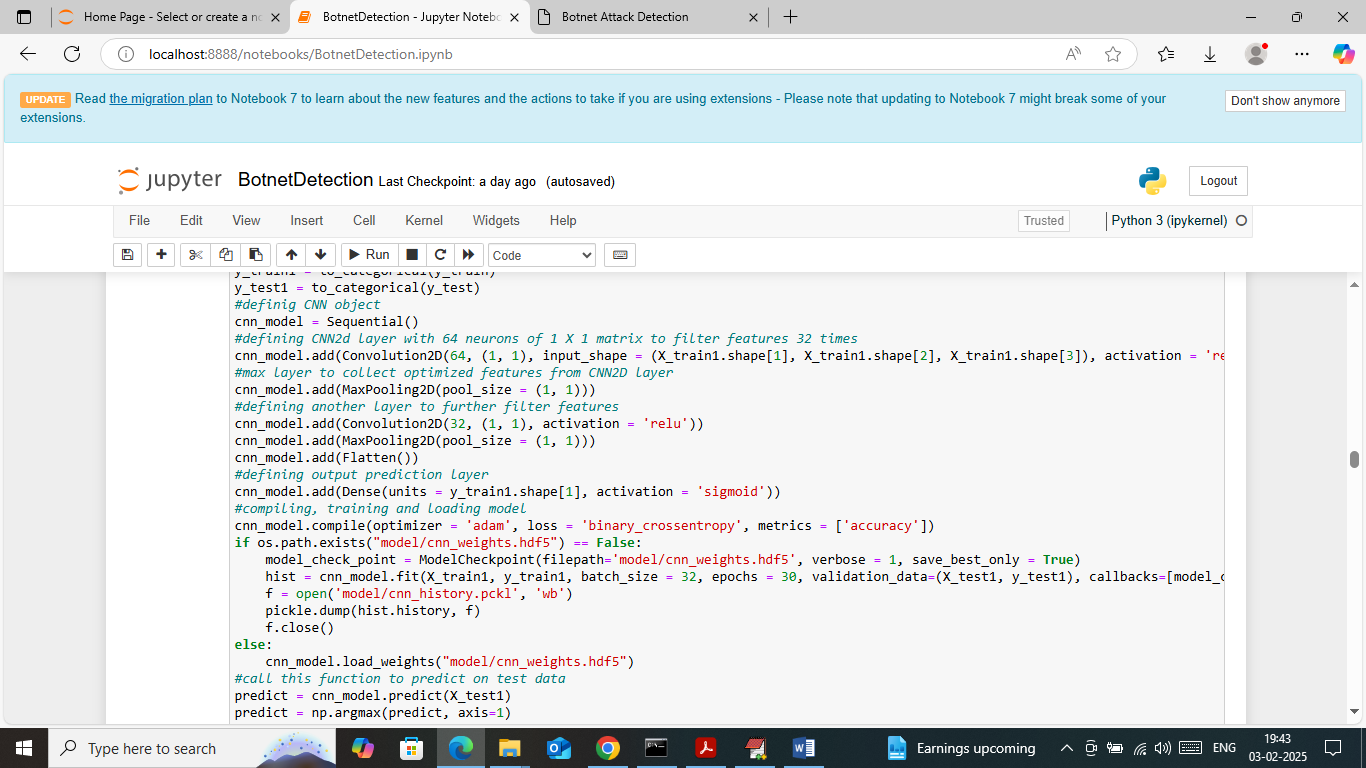
In above screen splitting dataset into train and test where application using 70% dataset for training and 30% for testing and then defining function to calculate accuracy and other metrics



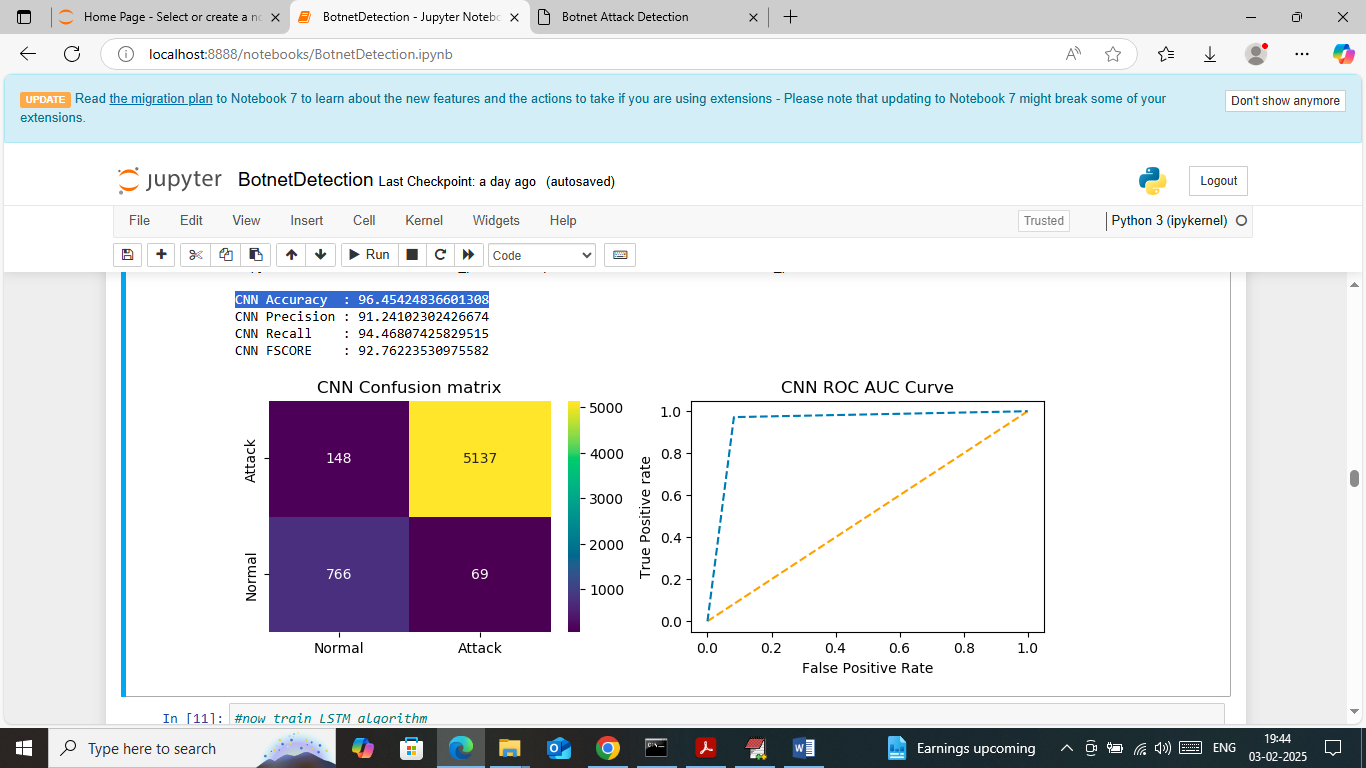
In above screen training ANN algorithm with different hyper parameters and after executing above block will get below output



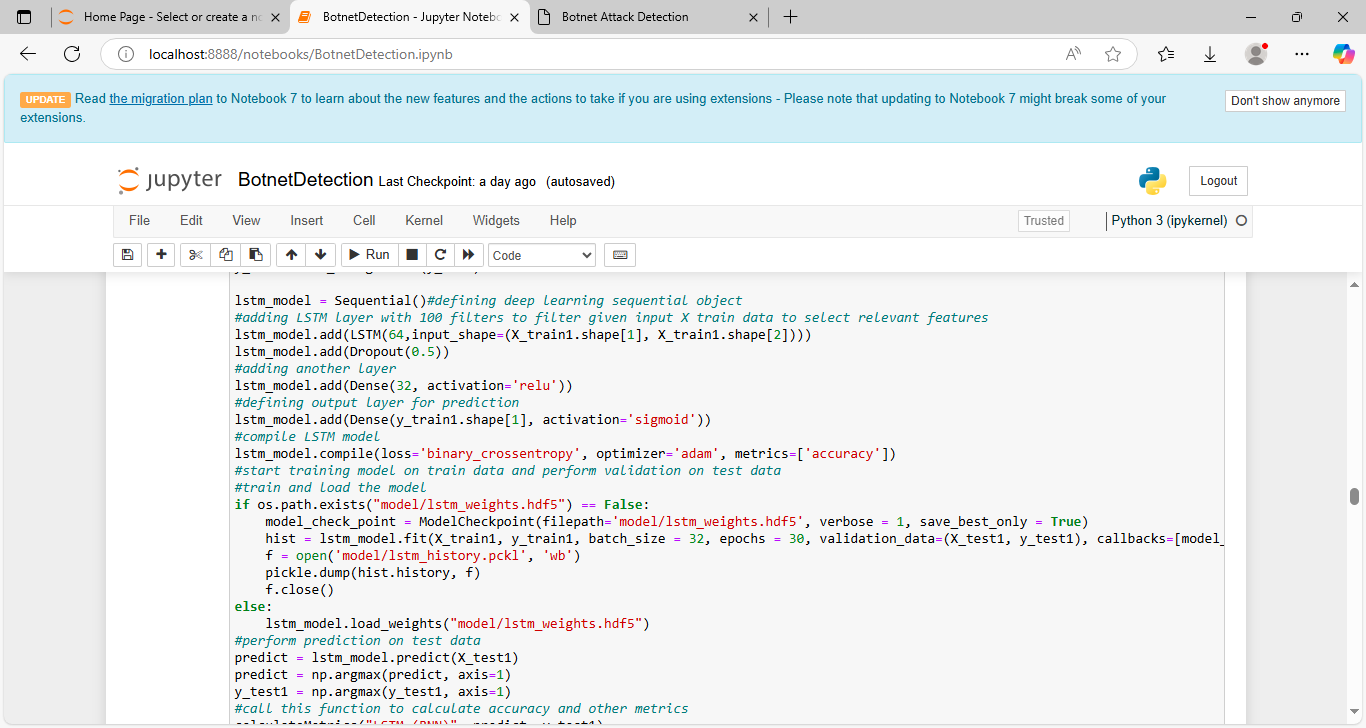
In above screen trained ANN algorithm on 70% training data and then performing prediction on 30% test data and then calculating prediction accuracy. In above screen ANN got 91% accuracy on test and can see other metrics like precision, recall and FSCORE. In above confusion matrix graph x-axis represents ‘Predicted Labels’ and y-axis represents True Labels and then light blue and yellow colour boxes in diagonal represents correct prediction count and remaining blue boxes represents incorrect prediction count which are very few. In ROC graph x-axis represents False Positive rate and y-axis represents True Positive rate and if blue line comes on top or orange line then all predictions are correct and if goes below orange line then predictions are incorrect



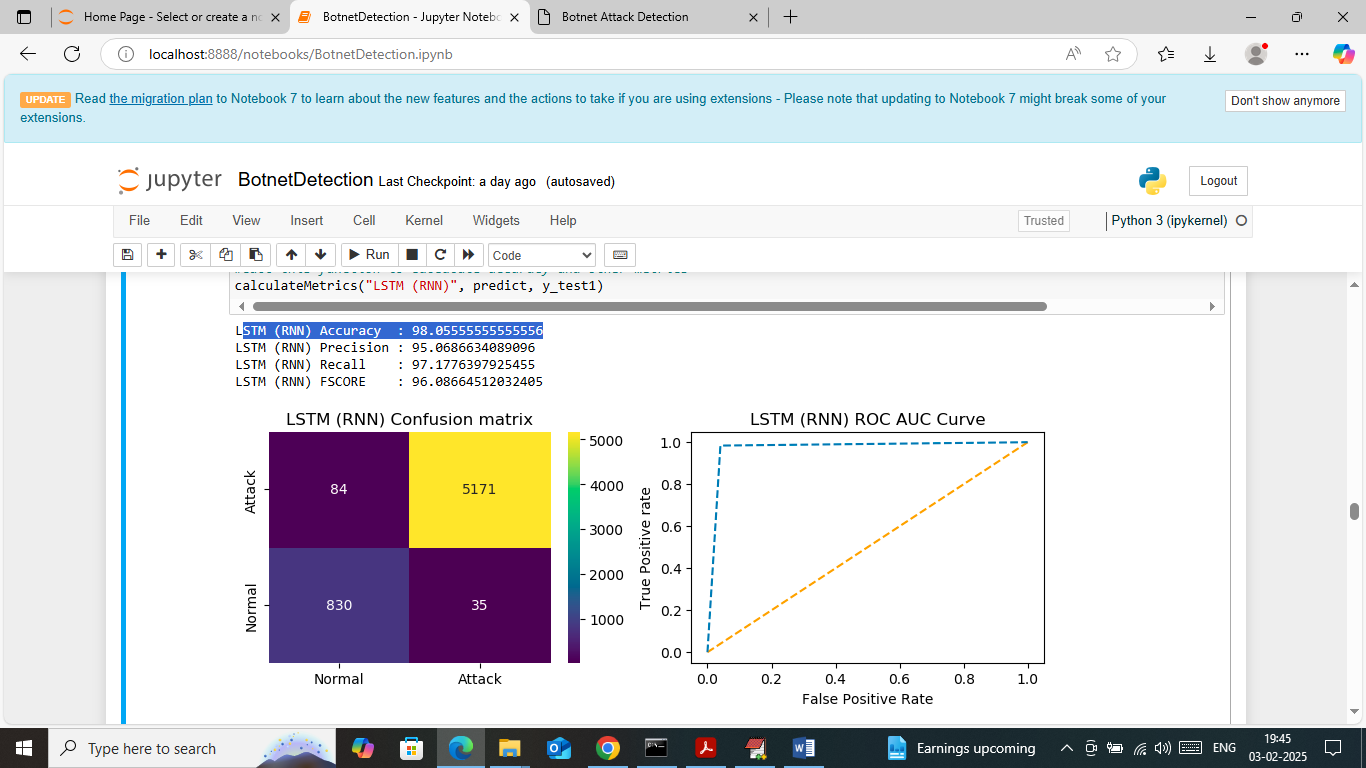
In above screen training CNN algorithm and below is the prediction output



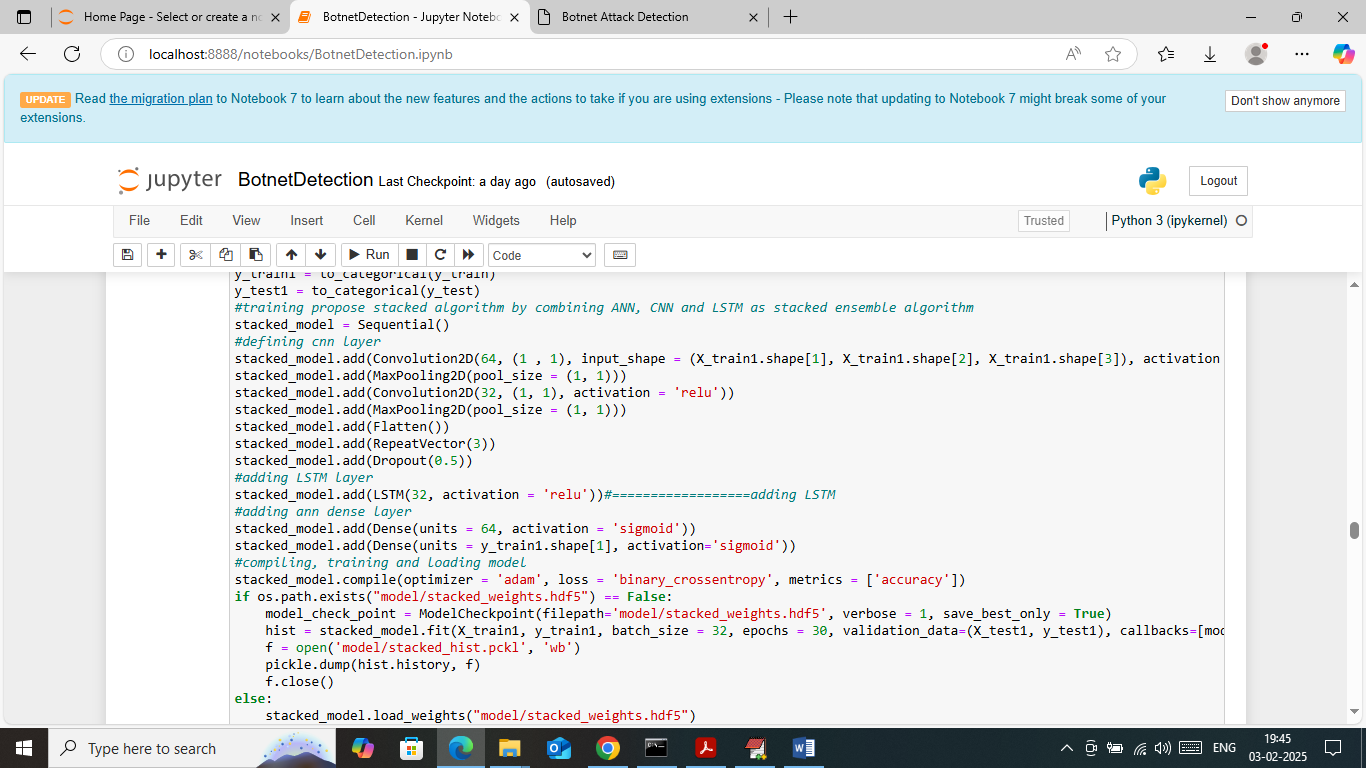
In above screen CNN got 96% accuracy and can see other metrics also



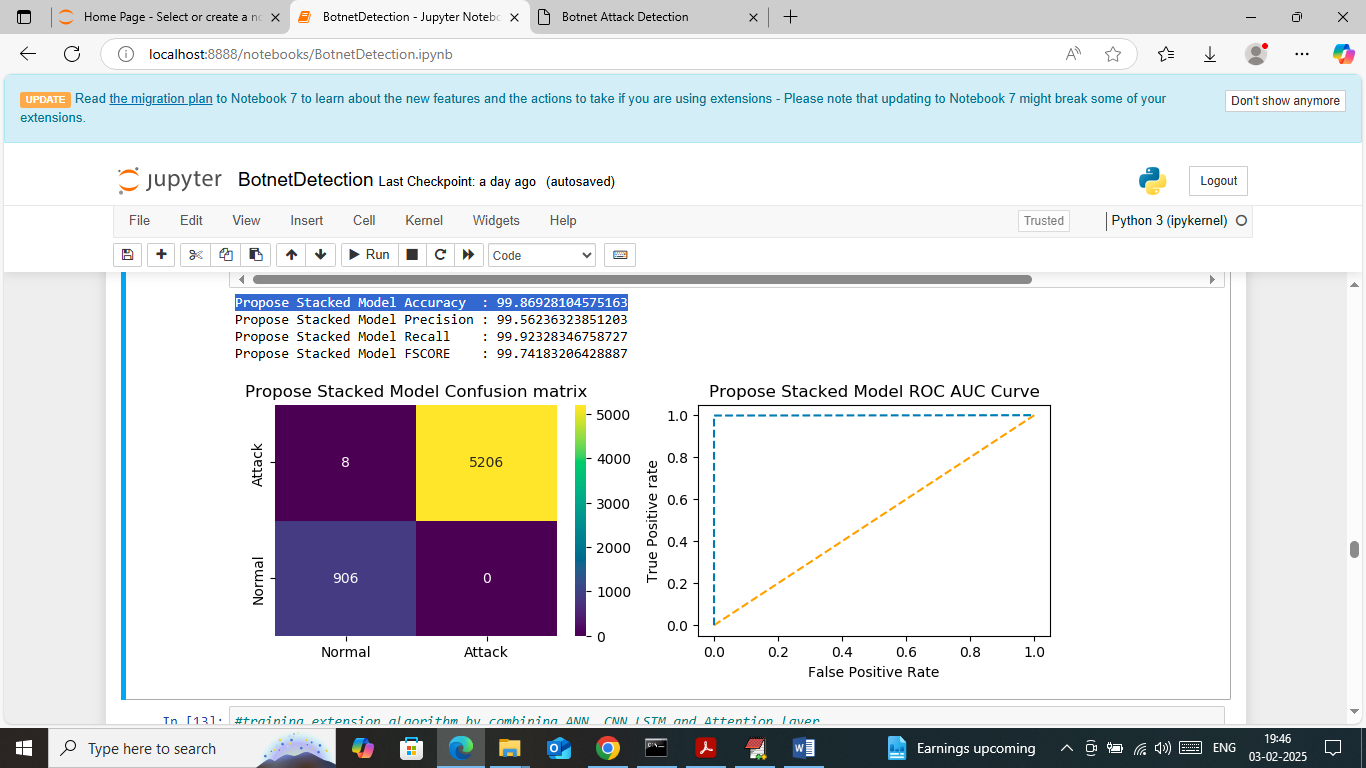
In above screen training LSTM RNN algorithm and below is the output



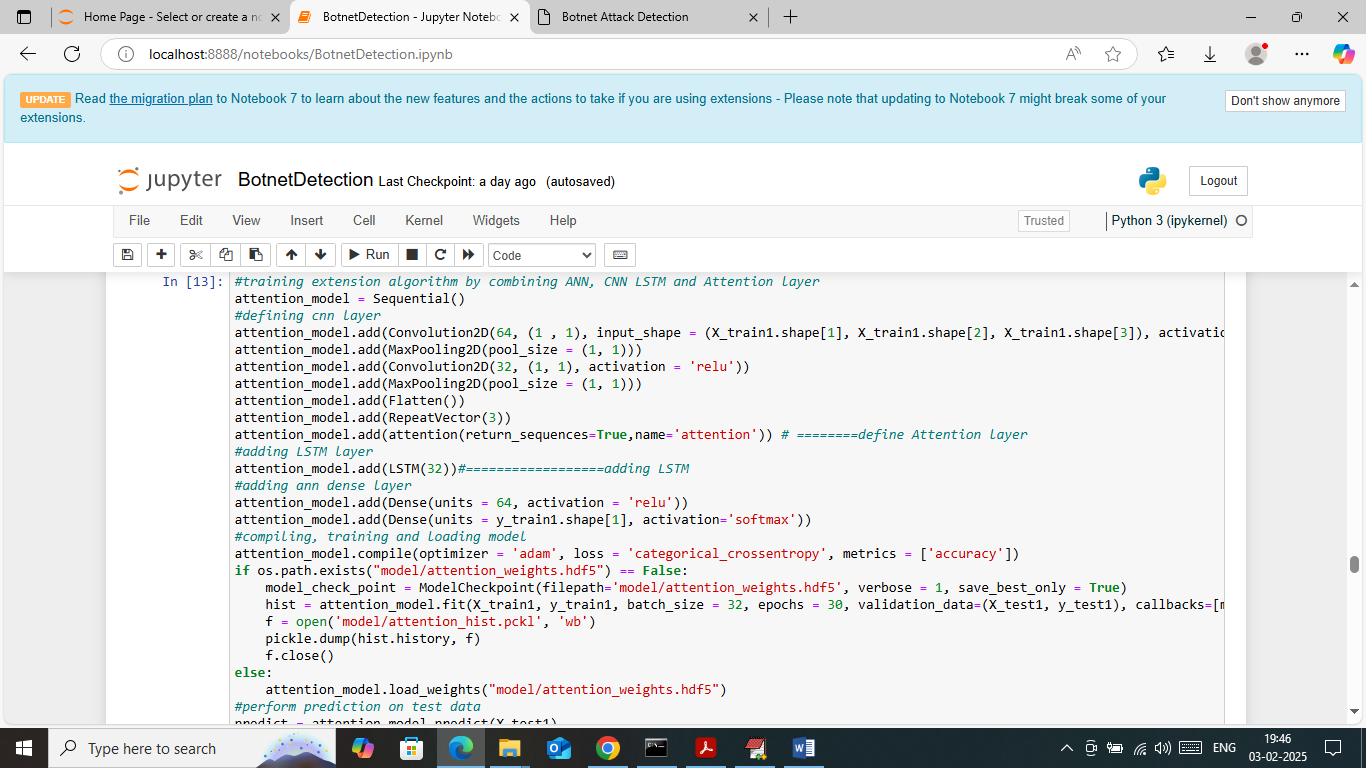
In above screen LSTM got 98% accuracy and can see other metrics also



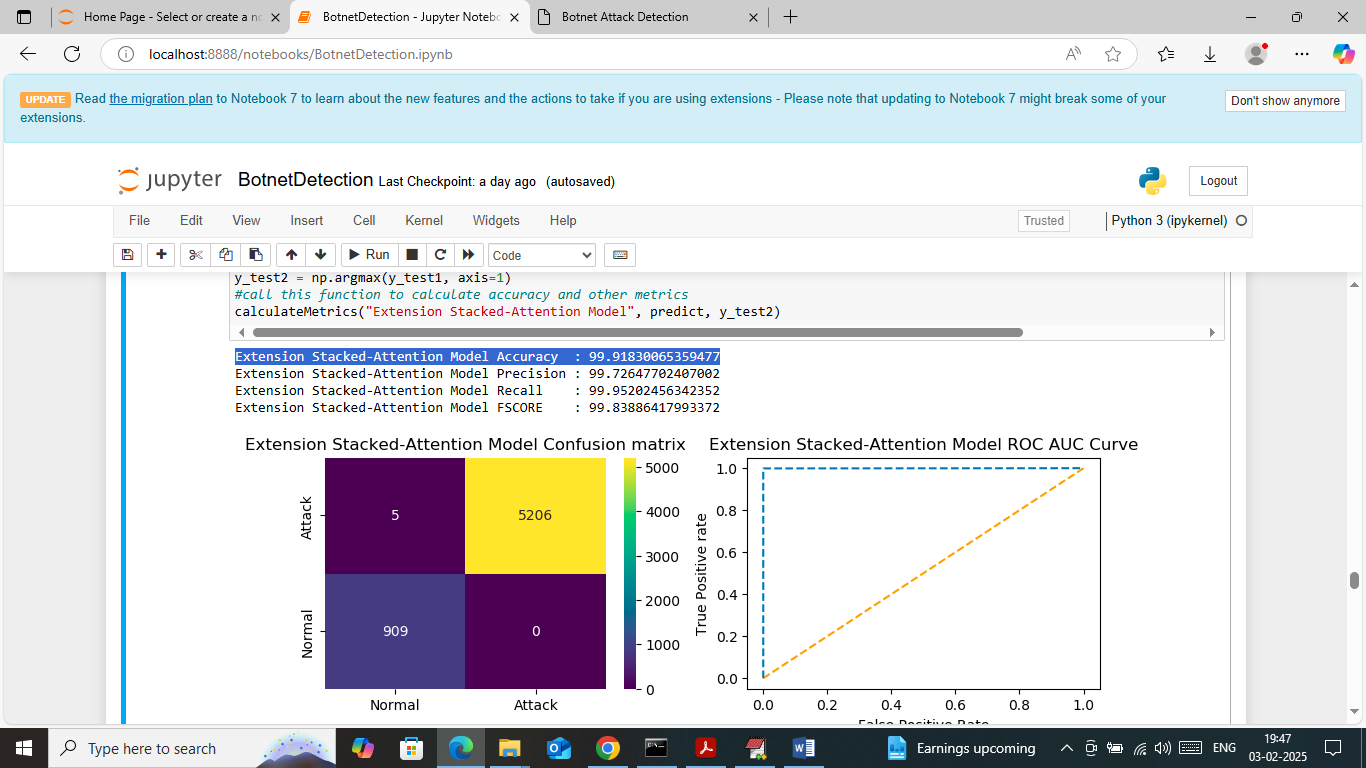
In above screen defining propose stacking algorithm by combining ANN + LSTM + RNN and CNN and after executing above block will get below output



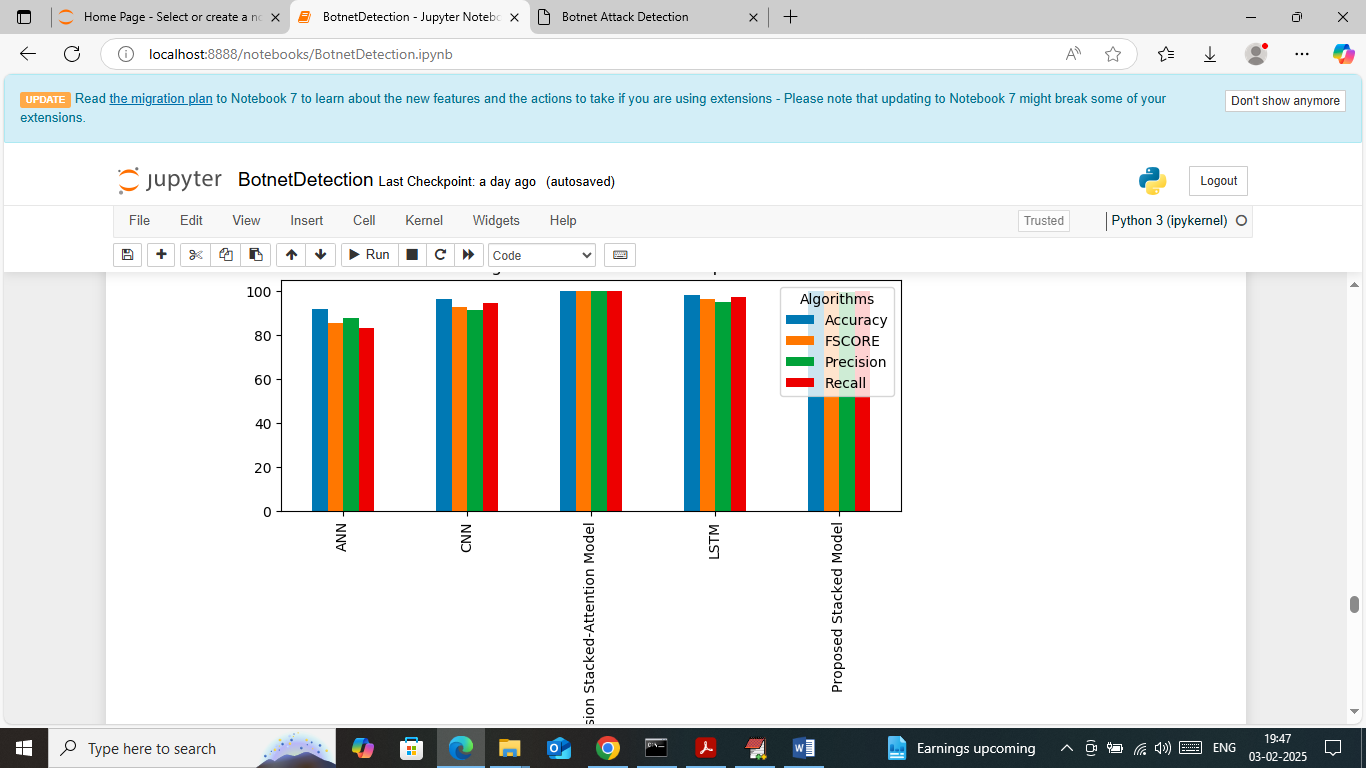
In above screen propose algorithm got 99.86% accuracy and can see other metrics also



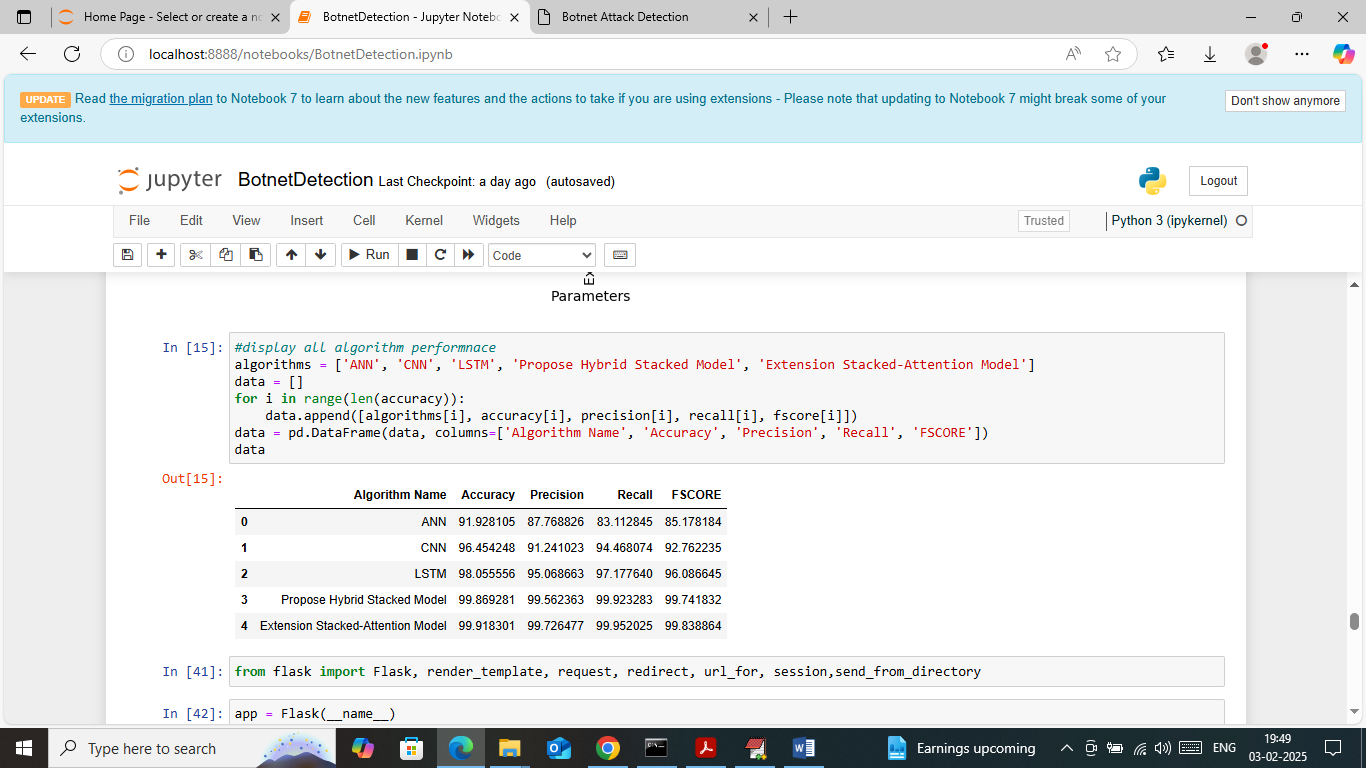
In above screen defining extension algorithm by combining propose Stacking (Hybrid) and attention layer and below is the output



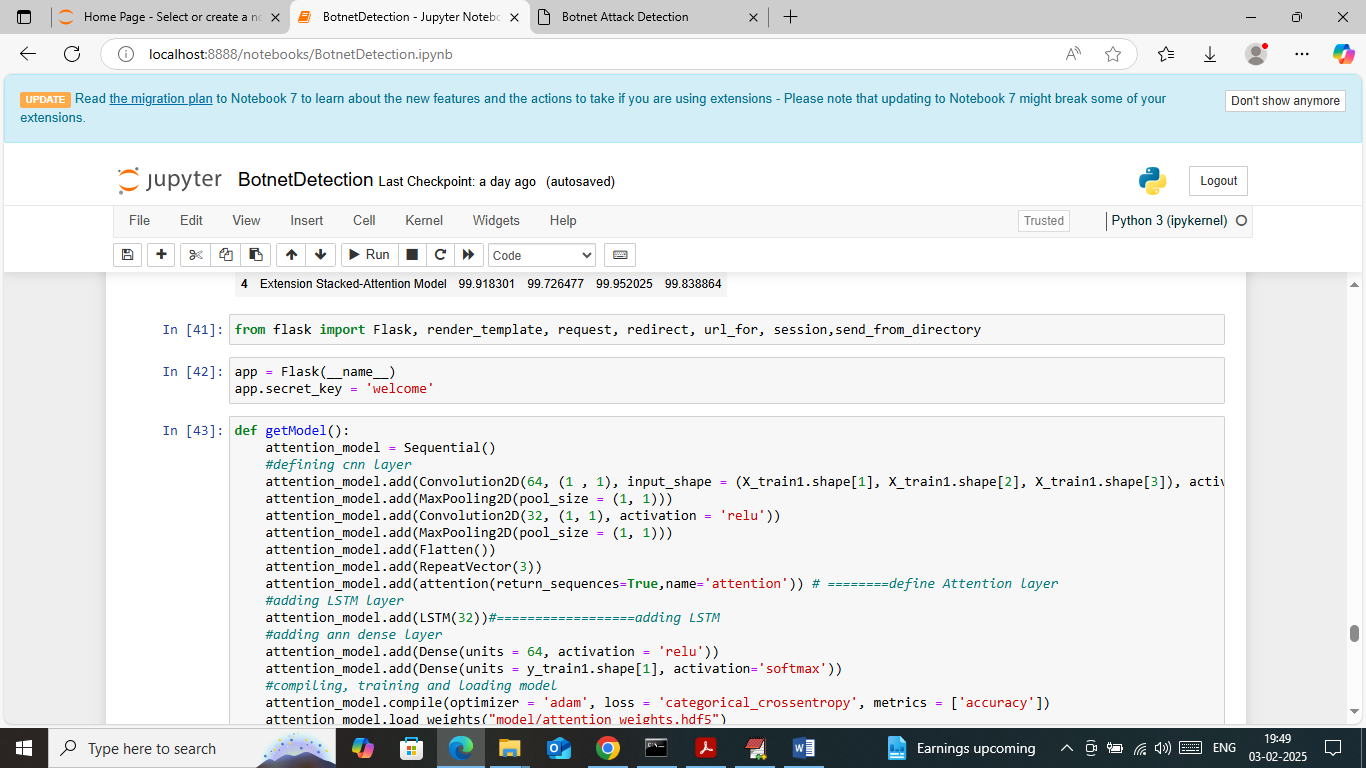
In above screen extension got 99.91% accuracy and can see other metrics also



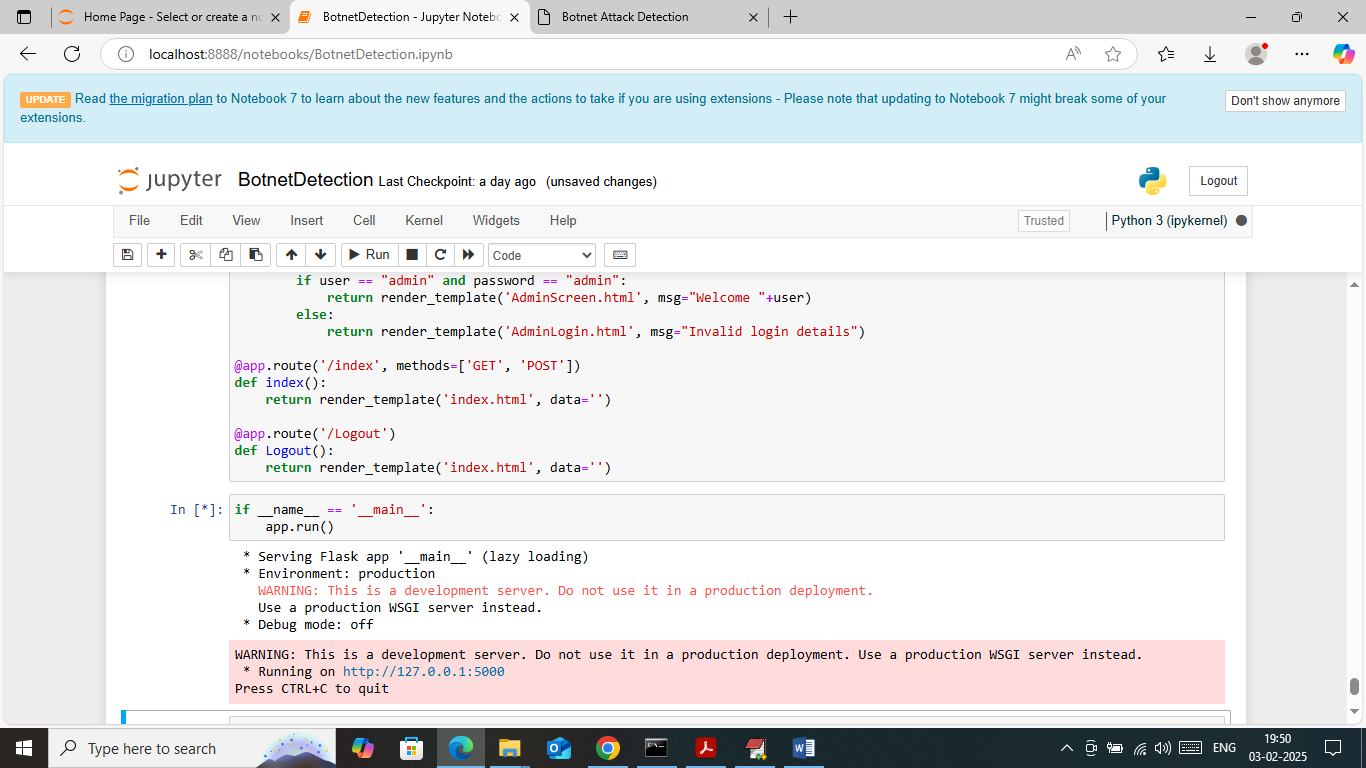
In above screen displaying comparison graph between all algorithms where x-axis represents algorithm names and y-axis represents accuracy and other metrics and in all algorithms propose and extension got high accuracy



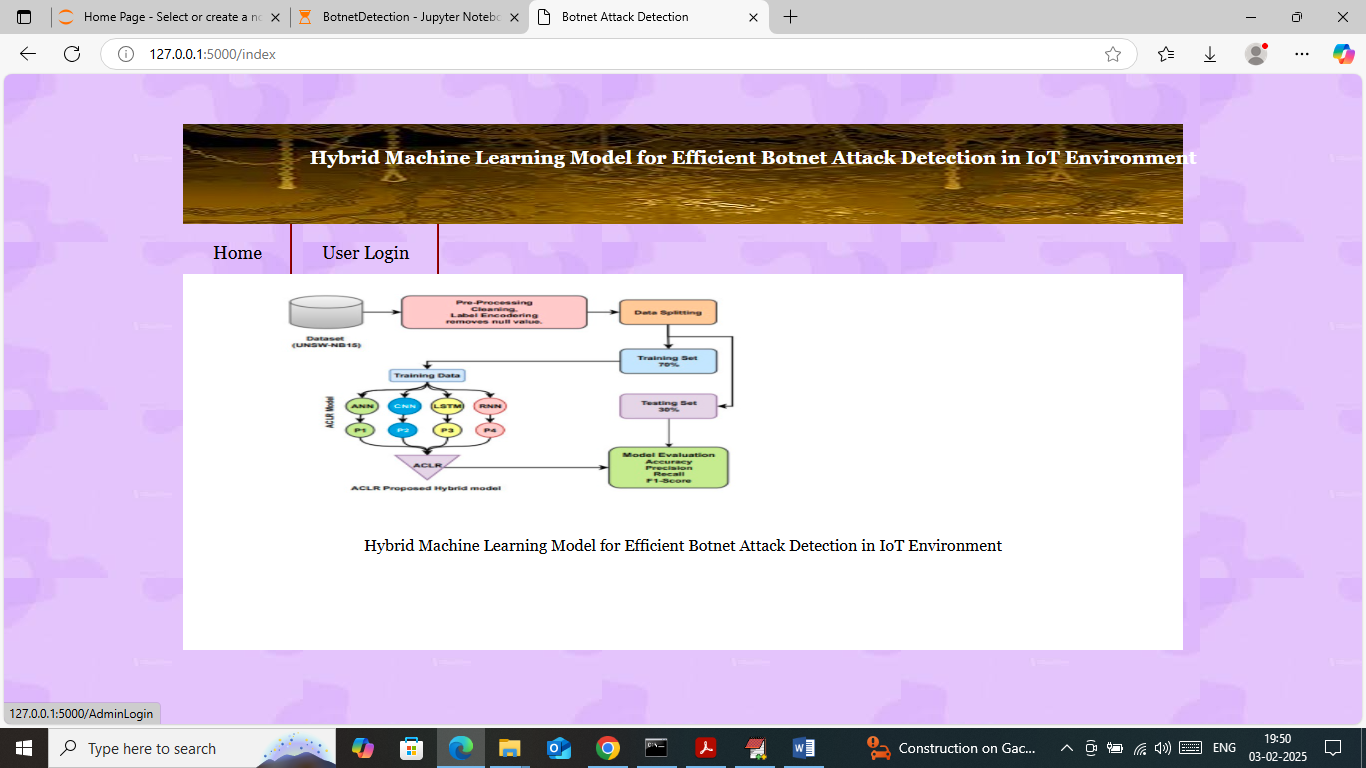
In above screen displaying all algorithm performance in tabular format



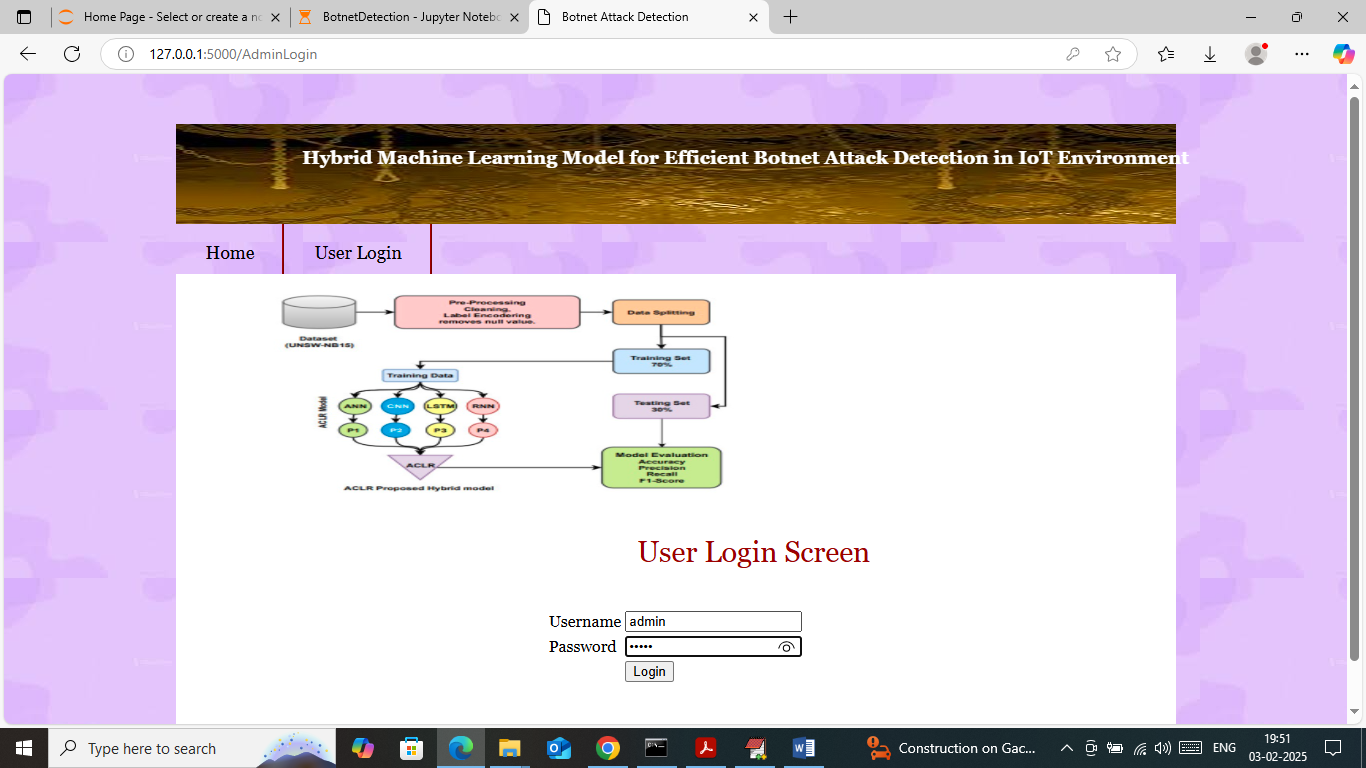
In above screen run all flask blocks code to start python web server and get below page



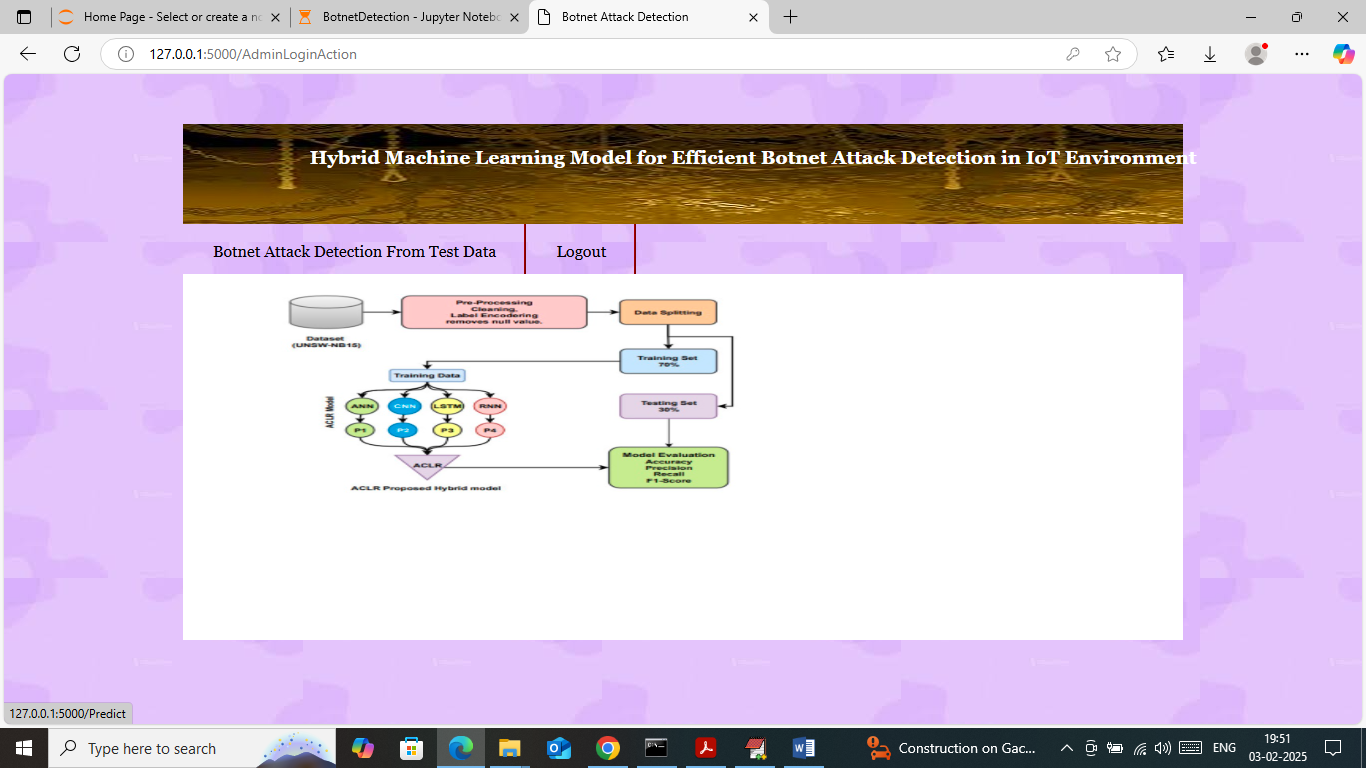
In above screen python flask server started and now open browser and enter URL as <http://127.0.0.1:5000/index> and then press enter key to get below page



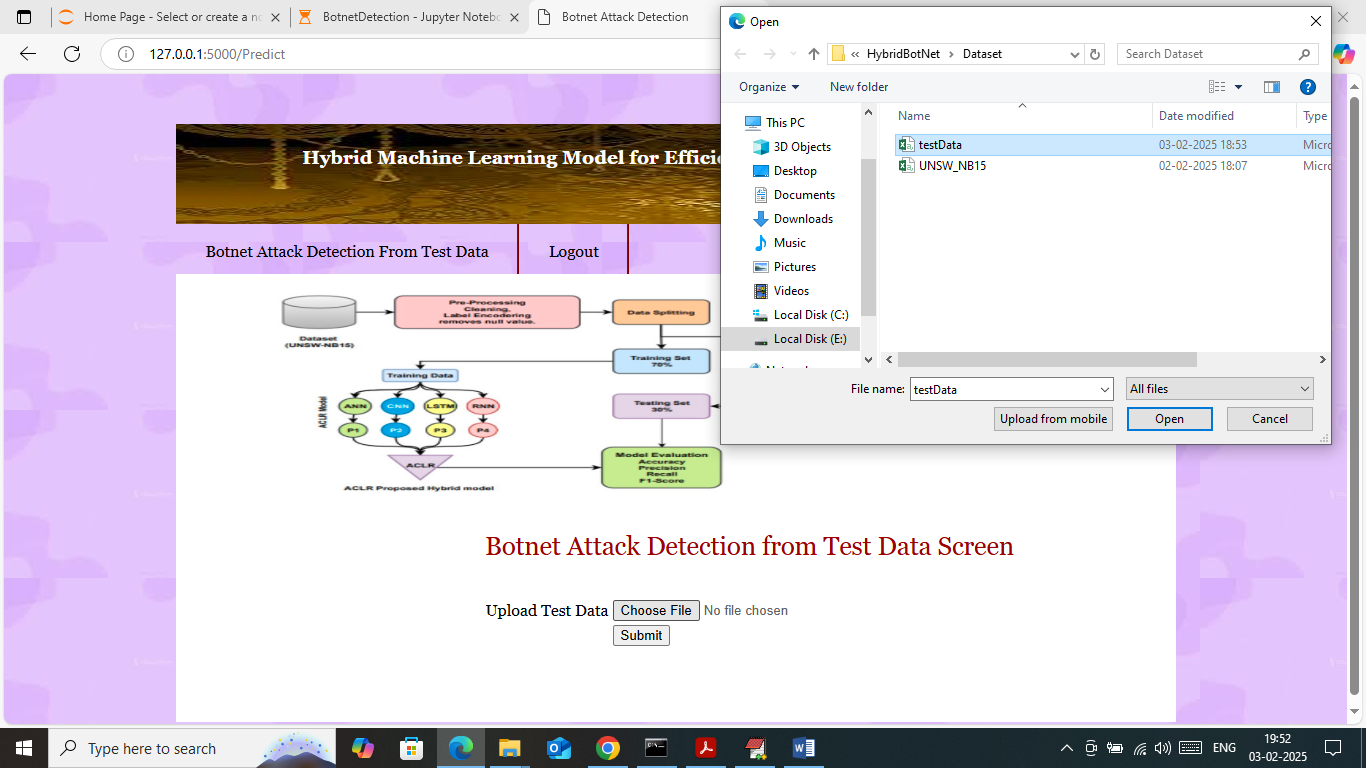
In above screen click on ‘User Login’ link to get below page



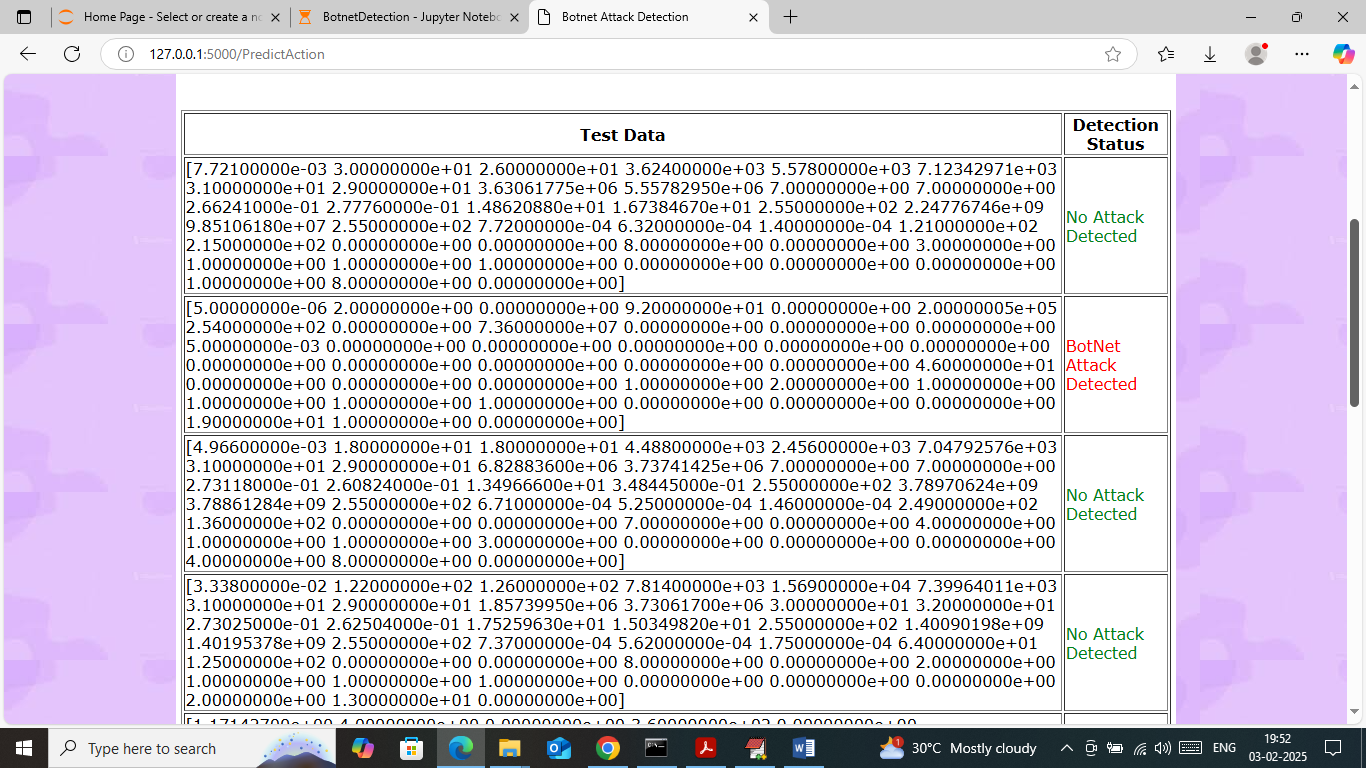
In above screen user can login by entering username and password as ‘admin and admin’ and after login will get below page



In above screen click on ‘Botnet Attack Detection’ link to get below page



In above screen select and upload ‘test Data.csv’ file from ‘Dataset’ folder and then click on all buttons to get below page



In above screen in first column can see TEST data values and in second column can see predicted output as ‘Attack or No Attack’.

Similarly by using above models you can detect attack from IoT environment.