Deep Learning for Secure Mobile Edge Computing in Cyber-Physical Transportation Systems

Mobile sensors or mobile devices are small electronic gadgets which runs on battery power and enable us to transfer files or data from one device to other device and mobile sensors sense its environment data and report to centralized server. This devices cannot perform heavy computation task like data encryption or image processing so Mobile Edge computing servers are introduced which will accept heavy computation task on behalf of mobile devices and then execute them and then send output back to mobile. All communications will happened between mobile devices and Edge servers through network. Sometime some malicious users will hack such network to perform malicious activities and to detect such malicious attacks author of this paper employing Deep Learning based Back-propagation DBN (deep belief network) algorithm.

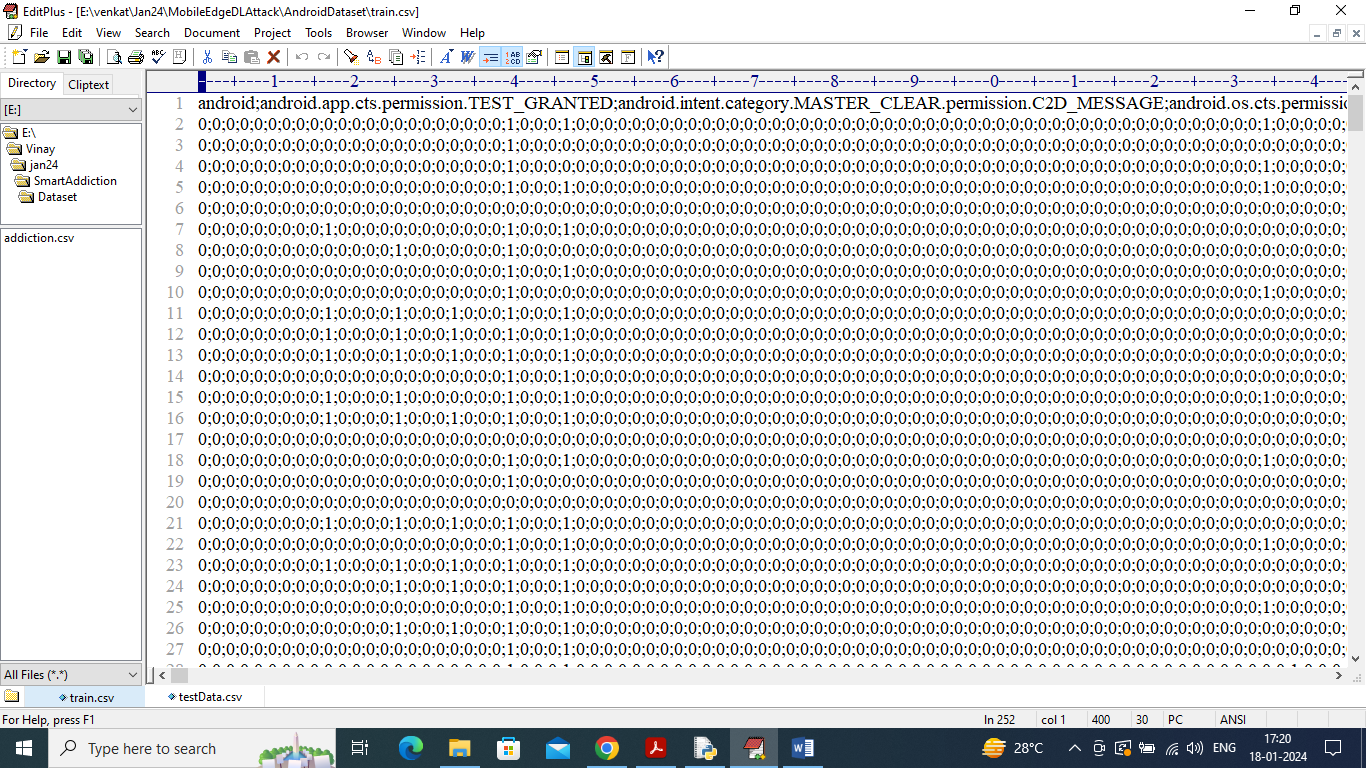
MEC is able to be used to execute the compute-intensive applications on the edge of transportation networks directly. As a result, the communications traffic is substantially increased among the connected edge devices. Therewith, communications security is emerging as a serious problem, and as an important research issue of the communications security, active feature learning is studied in propose work for actively detecting unknown attacks. A model based on deep learning is designed to learn attack features. This model uses unsupervised learning to accomplish the active learning process. In the evaluation, author has experimented with Android APK dataset. Propose algorithm is compare with other machine learning algorithms like SVM, Decision Tree and Random Forest. In all algorithms propose DBN algorithm is getting high detection accuracy.

Propose algorithm works I two stages

1. Features Processing Engine: in this stage android APK dataset will be given as input and then extract features from all APK dataset. This dataset contains both static and dynamic features where static features contains android Permission values and dynamic features include APK execution behaviour values
2. Attack Detection Features: in this stage DBN layers will be trained on input features in un-super-wise without any labels to differentiate dataset features as normal and attack and then Back-propagation layer will get trained with DBN selected features and attack labels to train a model which can detect attack from new un-label test data.

This deep belief network (DBN) combines a sequence of unsupervised networks (e.g., restricted Boltzmann machines, RBMs). An RBM is a kind of generative stochastic artificial neural network. In an RBM, there are a layer of visible units, a layer of hidden units, and corresponding weights between the visible unit’s layer and the hidden unit’s layer. Multiple RBMs can be stacked together, and the hidden layer output is the input of the next RBM. In our case, the reasonable number of hidden units, m, is 512. This value is obtained by adjusting with training.

To train propose and existing algorithm author has used Android APK dataset which is showing below



In above dataset screen first row contains dataset column names as Android Permissions and dynamic behaviour execution columns and remaining rows represents dataset values. In last column we have class label as 0 and 1 where 0 means ‘Non-Attack’ and 1 means ‘Attack’.

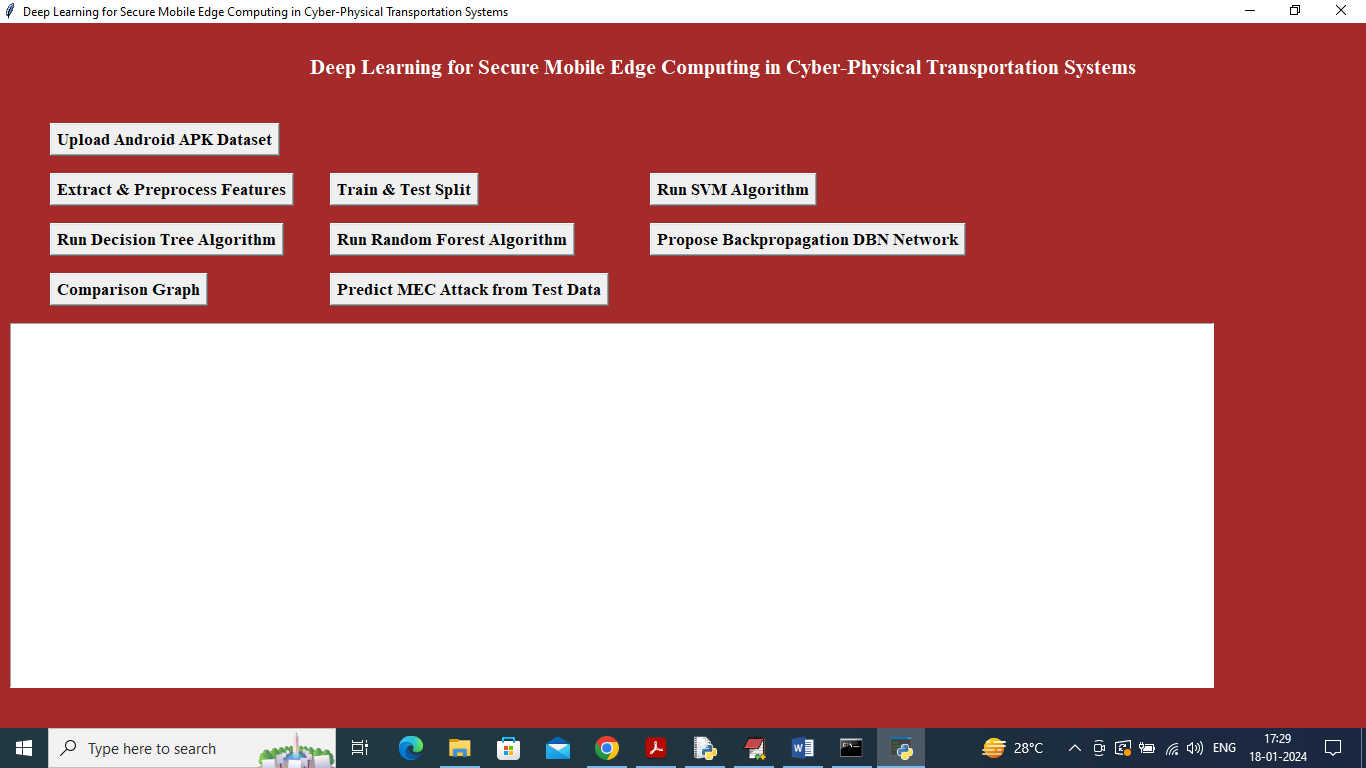
So by using above dataset we will train and test all existing and propose algorithms. Each algorithm performance is evaluated in terms of accuracy, precision, recall, Confusion Matrix Graph, ROC graph and FSCORE.

To implement this project we have designed following modules

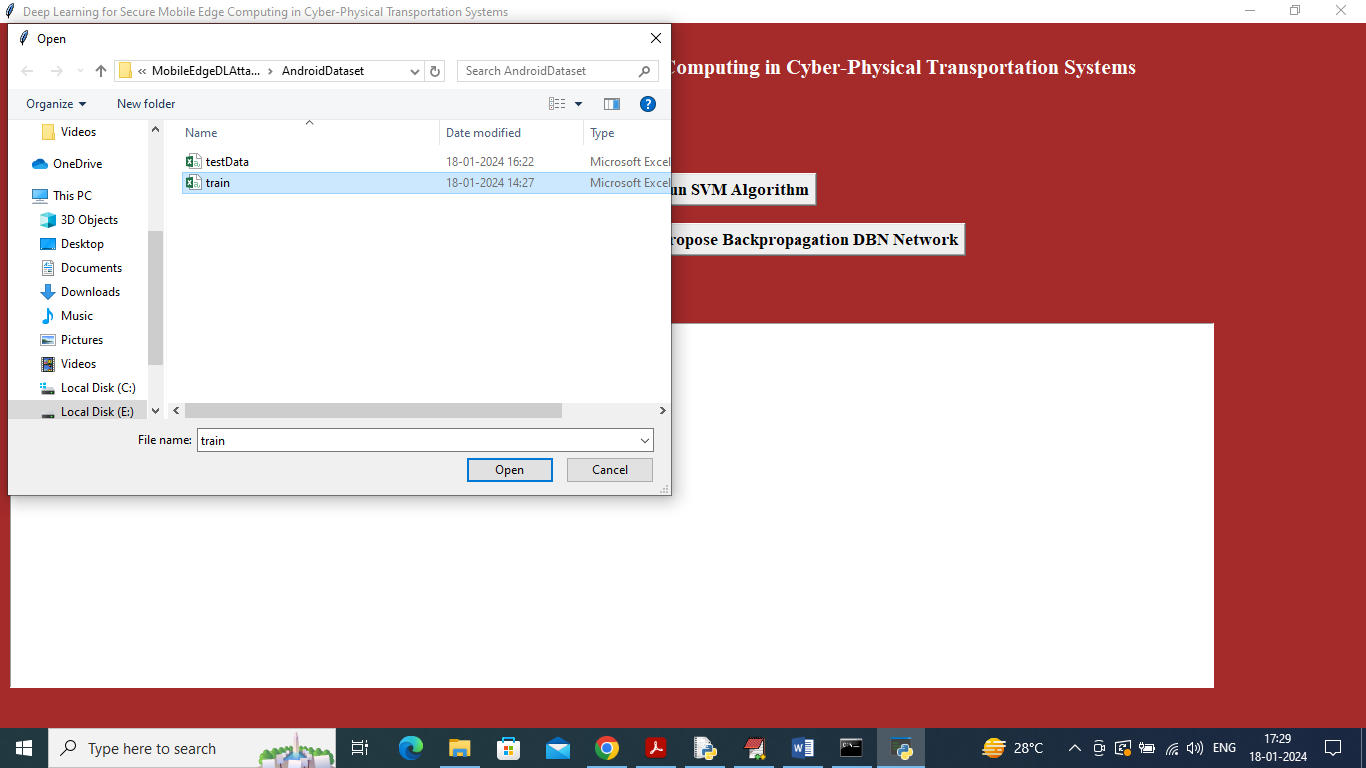
1. Upload Android APK Dataset: using this module we will upload Android APK dataset to application and then read all dataset values and then plot graph of normal and attack records
2. Extract & Pre-process Features: from loaded dataset features will extracted and then normalize, shuffle and replace missing values with 0
3. Train & Test Split: split dataset into train and test where application using 80% dataset for training and 20% for testing
4. Run SVM Algorithm: 80% training features dataset will be input to SVM algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
5. Run Decision Tree Algorithm: 80% training features dataset will be input to Decision Tree algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
6. Run Random Forest Algorithm: 80% training features dataset will be input to Random Forest algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
7. Propose Back-propagation DBN Network: 80% training features dataset will be input to ‘Propose Back-propagation DBN Network’ algorithm to train a model and this model will be applied on 20% test data to calculate prediction accuracy
8. Comparison Graph: will plot comparison graph between all algorithms
9. Predict MEC Attack from Test Data: using this module will upload test data and then Propose DBN algorithm will analyse all test data to predict weather test data is normal or contains attack behaviour

SCREEN SHOTS

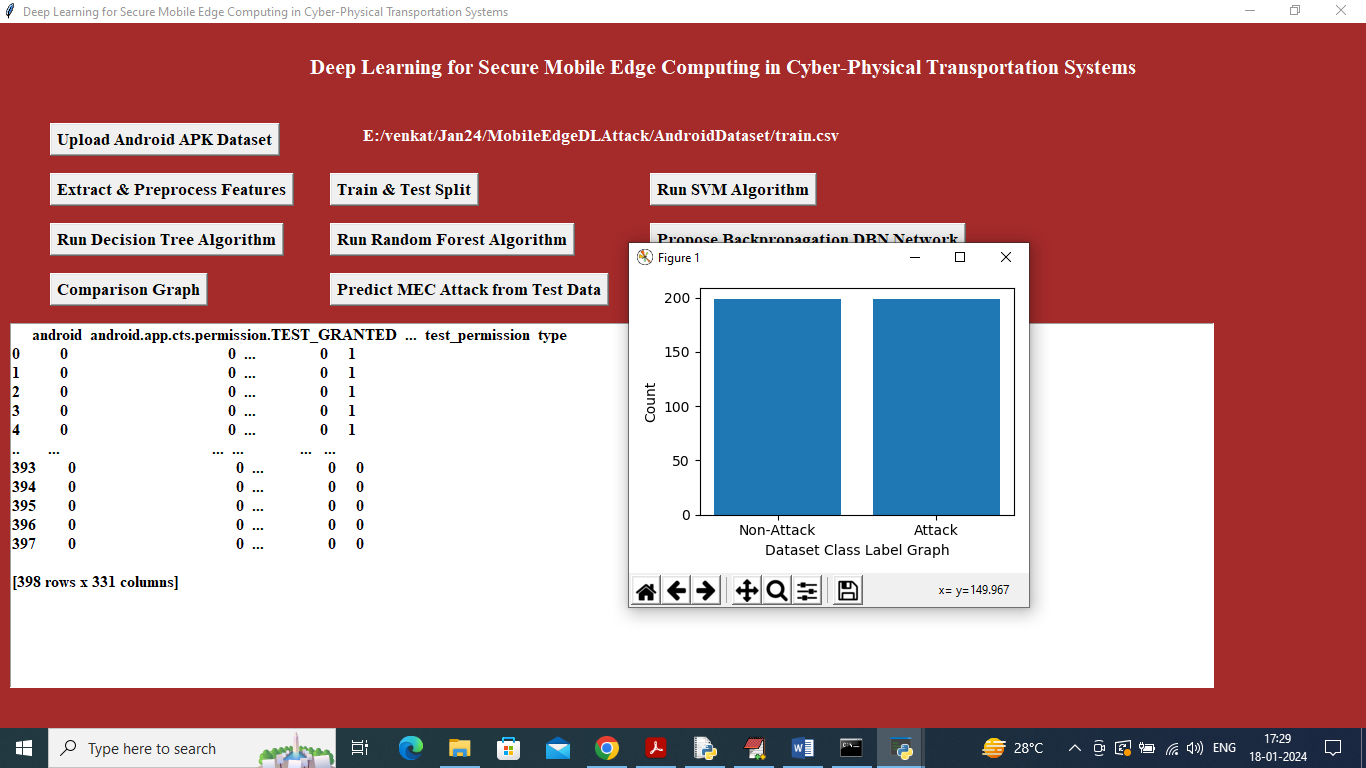
To run project double click on ‘run.bat’ file to get below screen



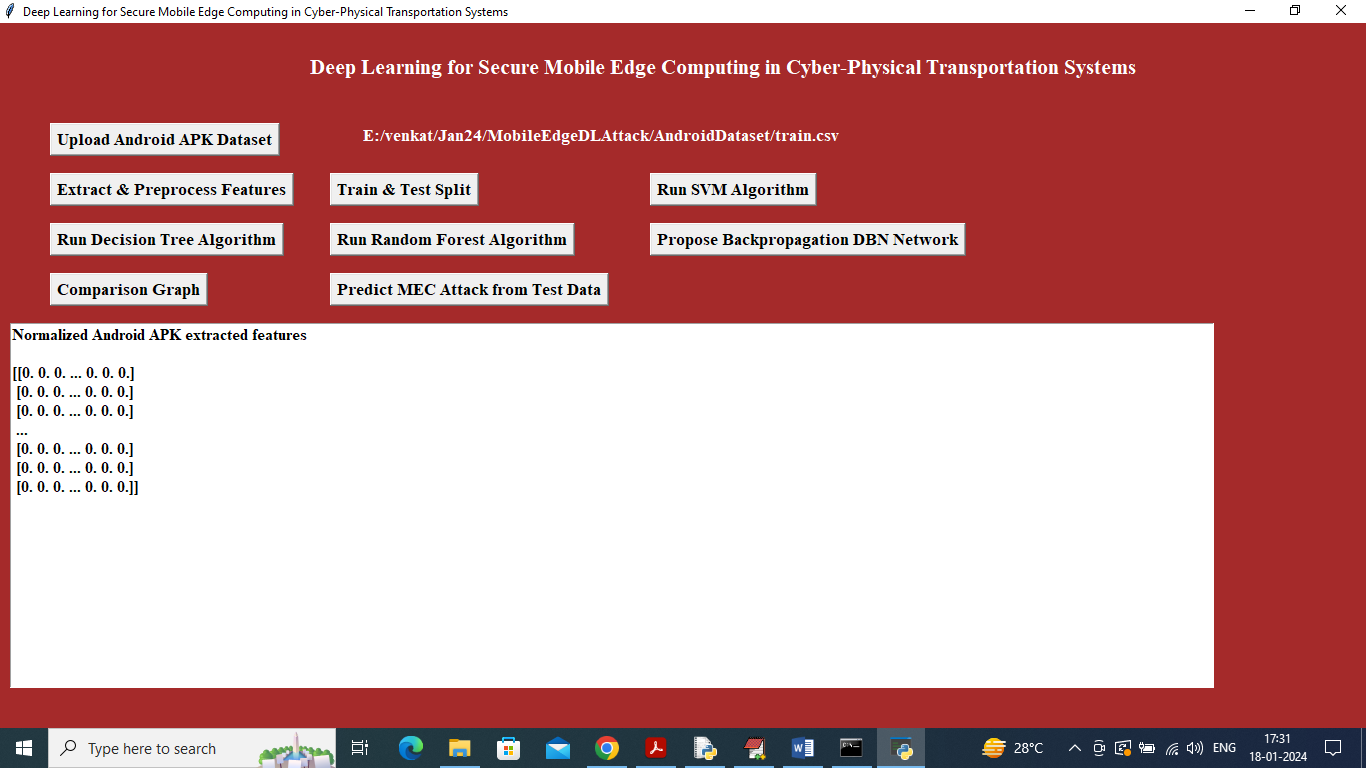
In above screen click on ‘Upload Android APK Dataset’ button to upload dataset and then will get below output



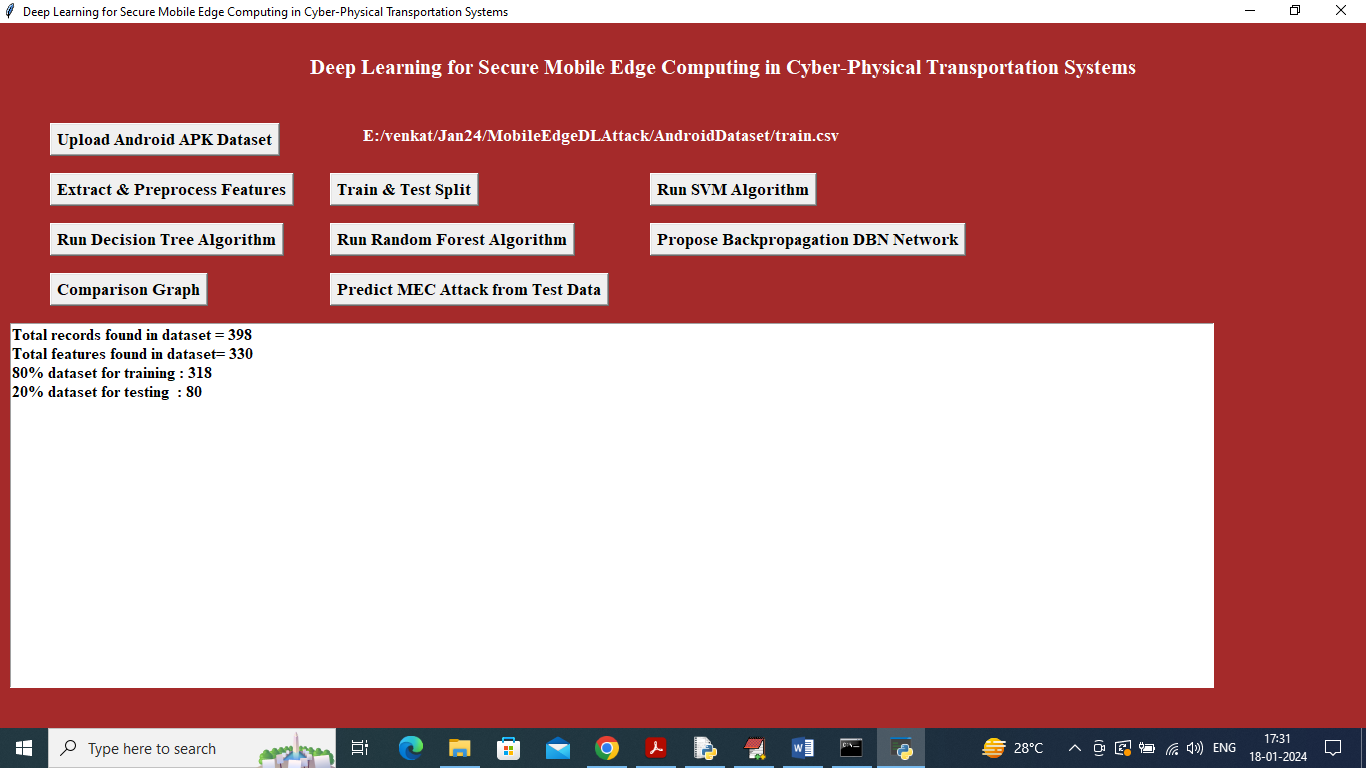
In above screen selecting and uploading ‘train’ dataset file and then click on ‘Open’ button to load dataset and then will get below output



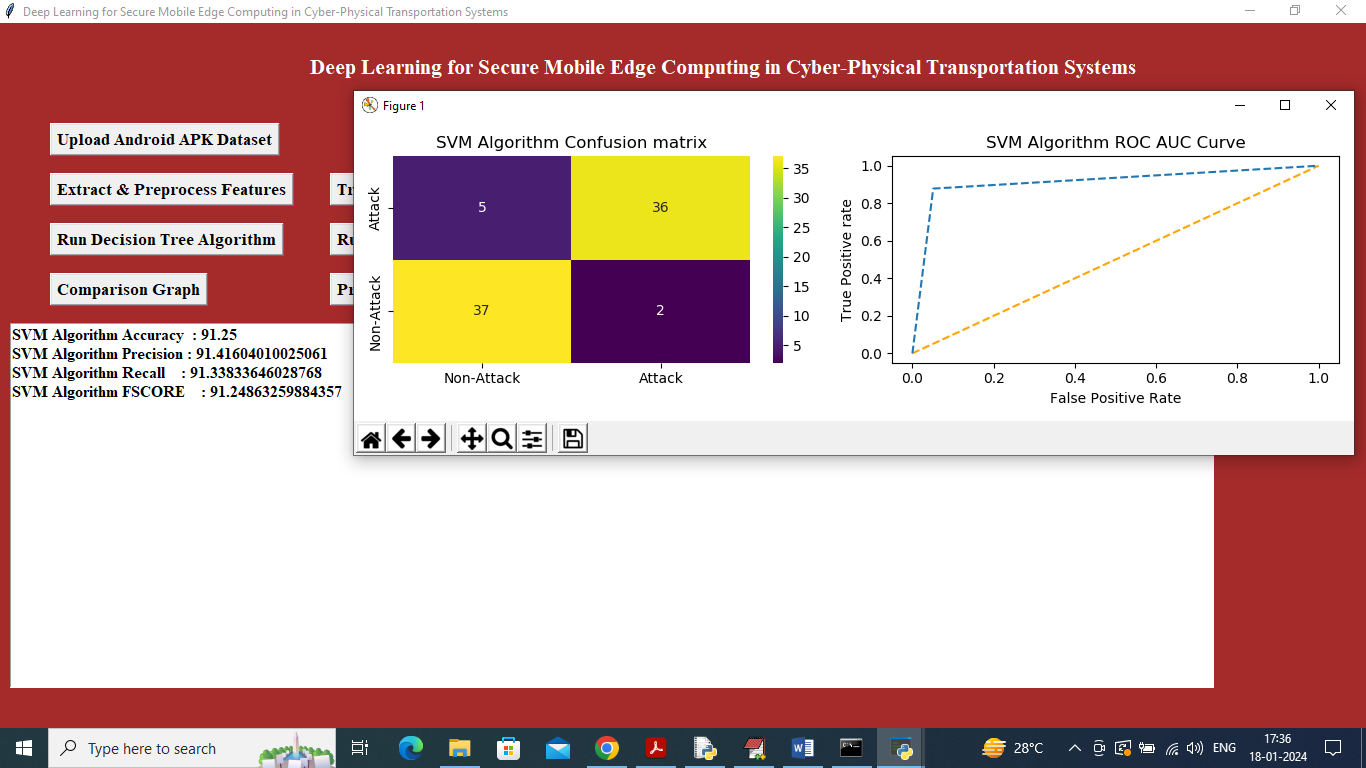
In above screen Android APK permission and behaviour dynamic dataset loaded and in graph x-axis represents class label as ‘Attack or Non-Attack’ and y-axis represents number of records and now close above graph and then click on ‘Extract & Pre-process Features’ button to extract and process features



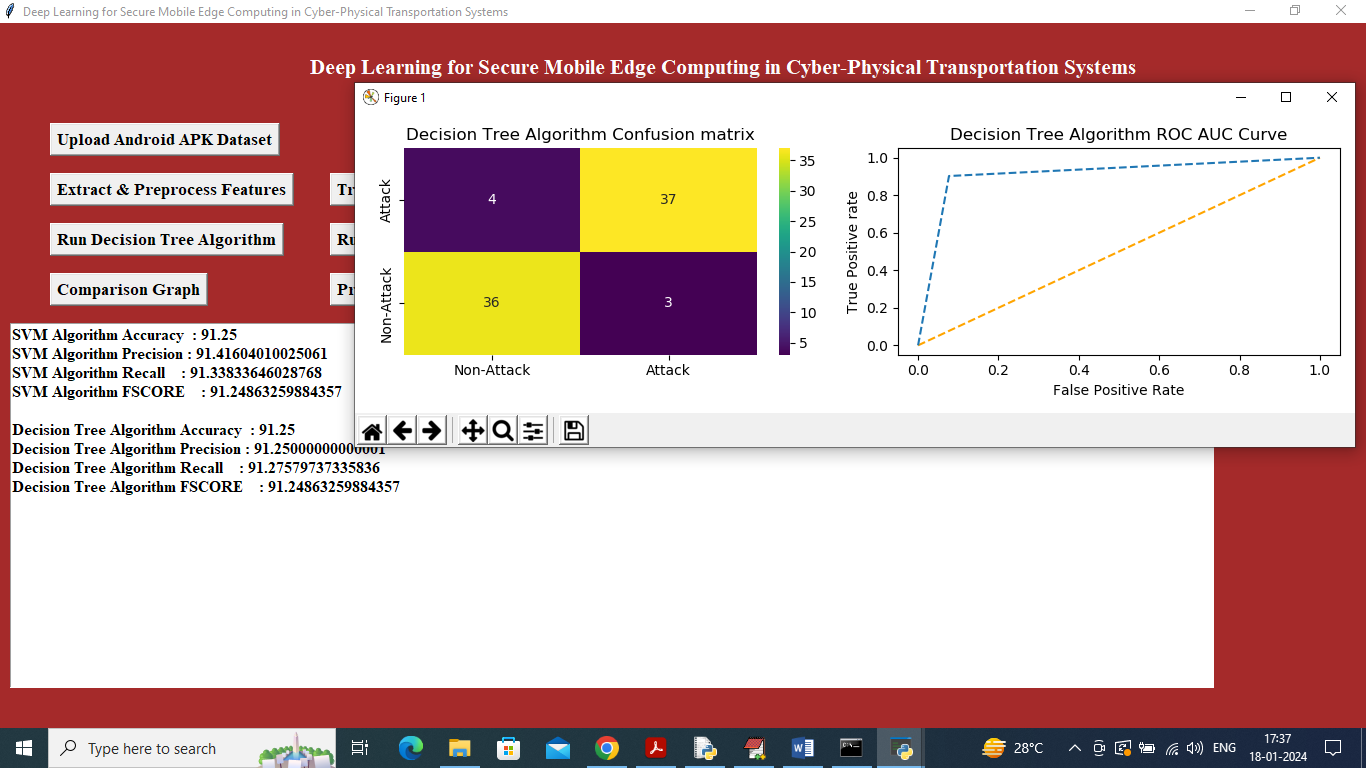
In above screen features processing completed and now click on ‘Train & Test Split’ button to split dataset and then will get below output



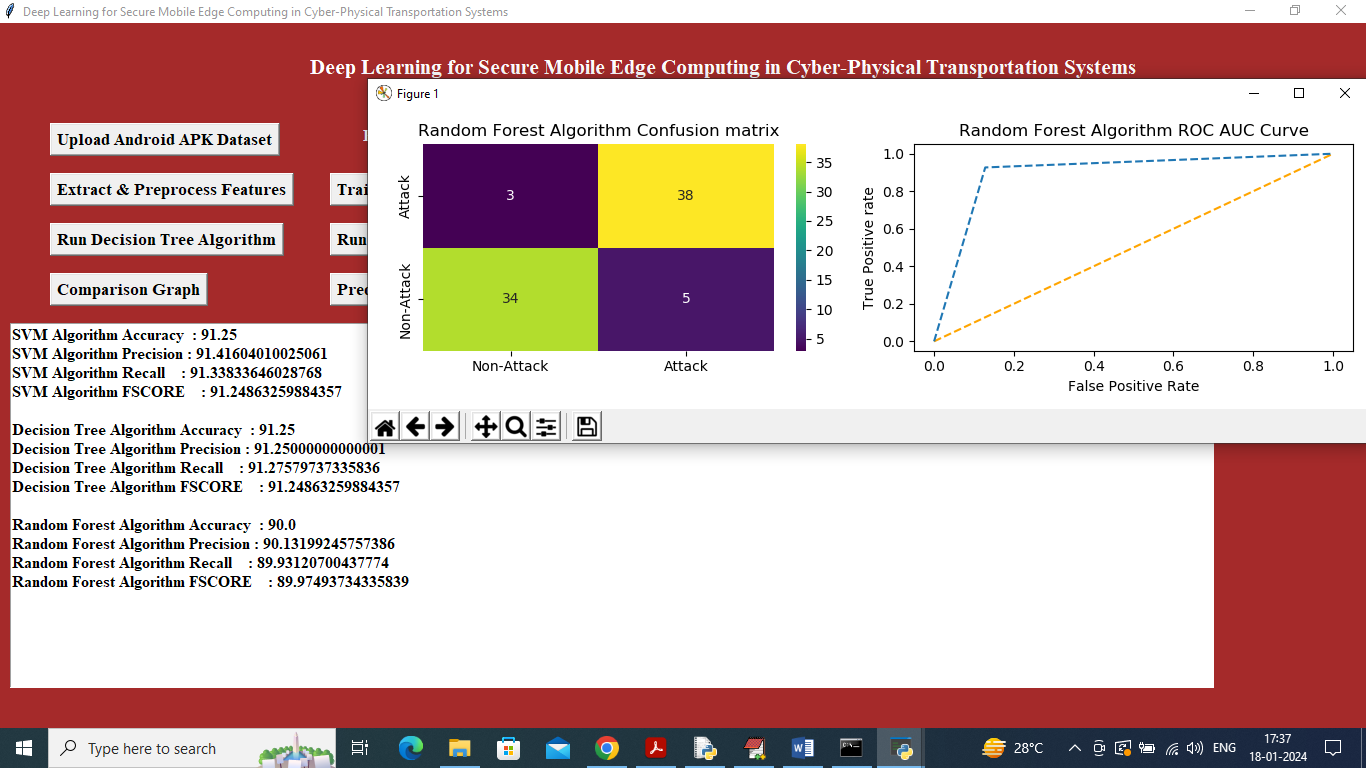
In above screen can see total records with total features in each record and then can see train and test size and now click on ‘Run SVM Algorithm’ button to train SVM and then will get below output



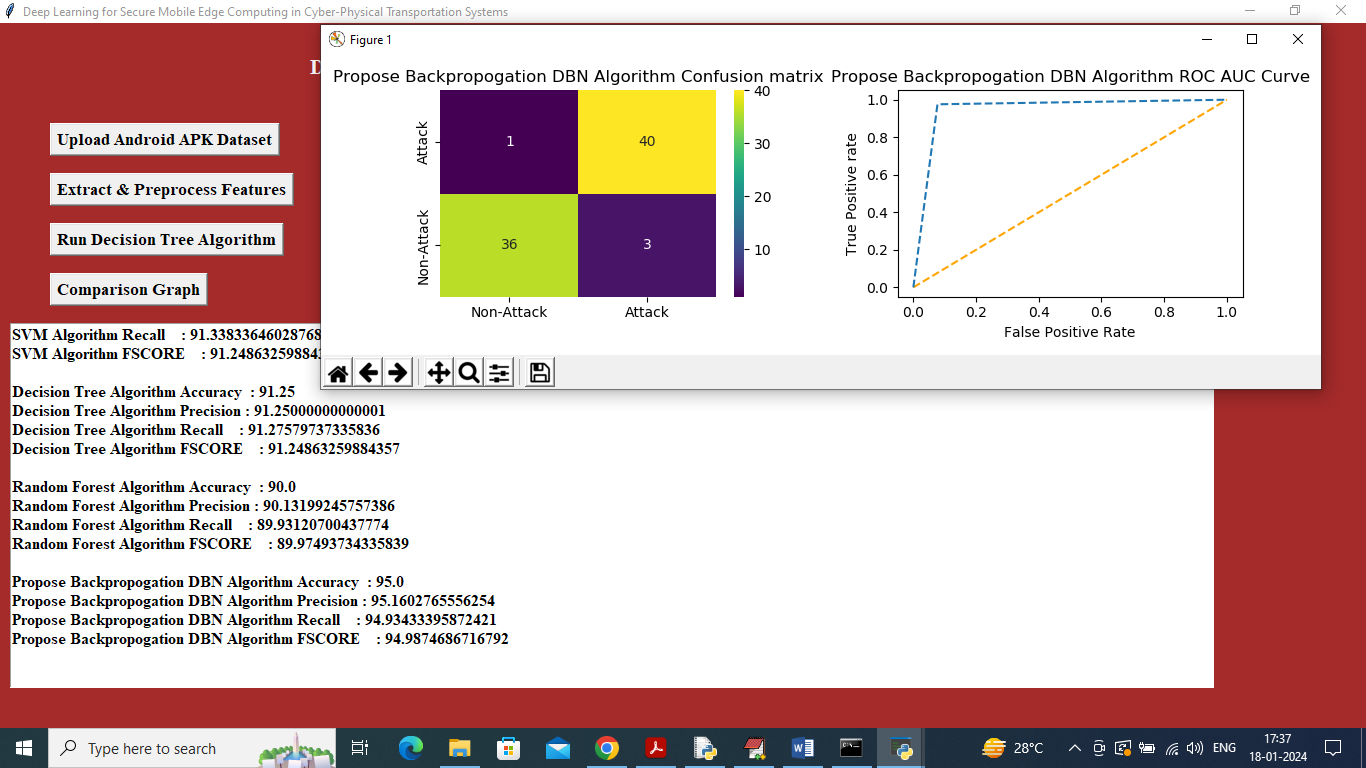
In above screen training SVM algorithm and then it got 91% accuracy and can see other metrics like precision, recall and etc. in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True labels and then yellow boxes in diagnol represents correct prediction count and all blue 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 lines comes below orange line then all predictions are incorrect or false and if goes above orange line then all predictions are correct or true. Now close above graph and then click on ‘Run Decision Tree’ button to train decision tree and get below output



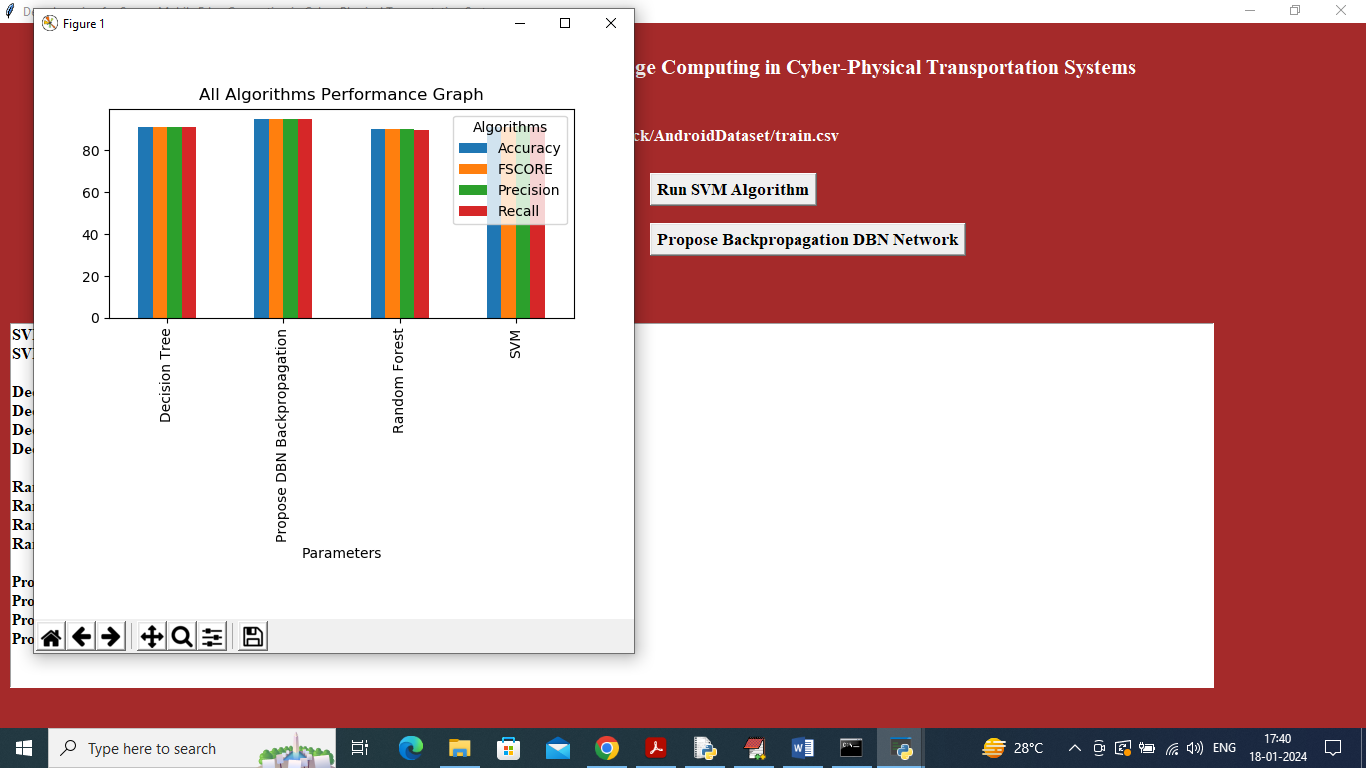
In above screen decision Tree got 91% accuracy and can see other metrics output and now close above graph and then click on ‘Run Random Forest’ button to get below output



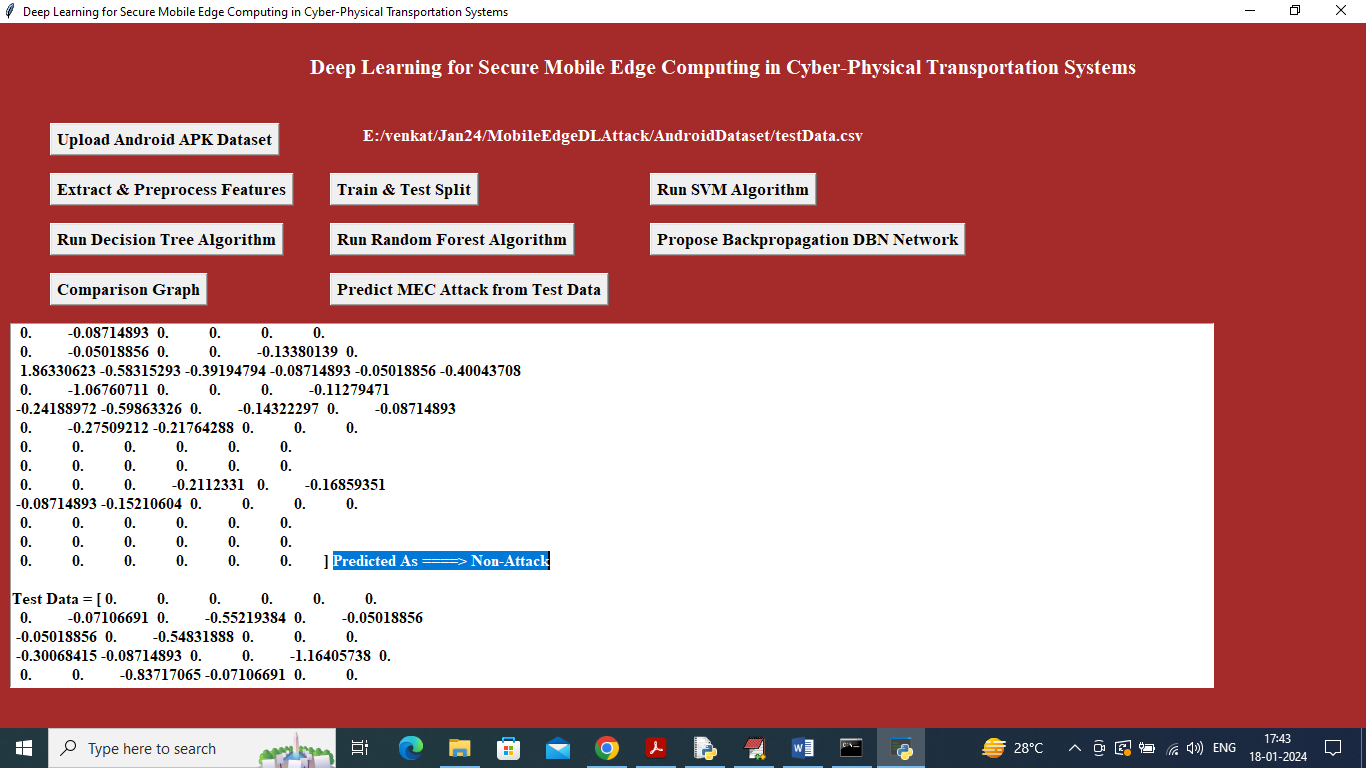
In above screen Random Forest got 90% accuracy and can see other metrics output and now close above graph and then click on ‘Propose Back-propagation DBN Network’ button to get below output



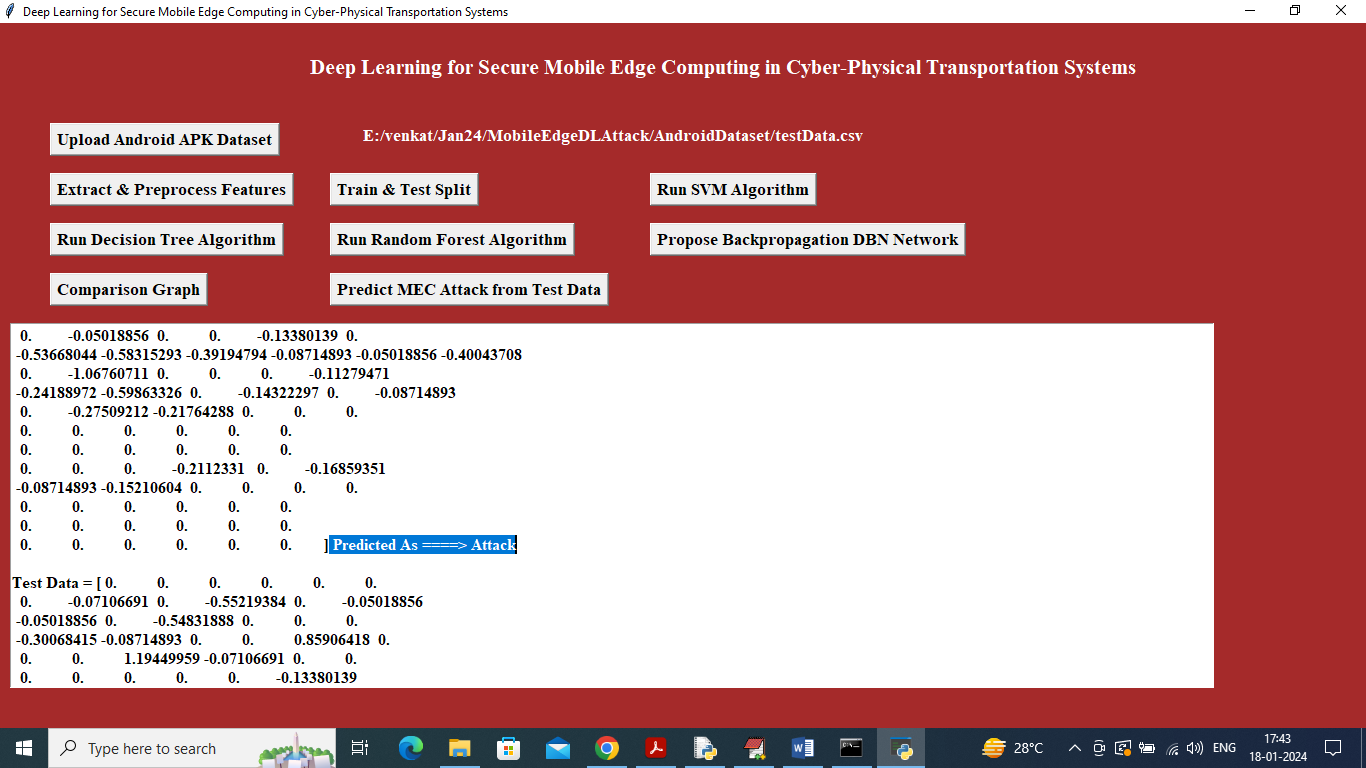
In above screen ‘Propose Back-propagation DBN Network’ got 95% accuracy and can see other metrics output and now close above graph and then click on ‘Comparison Graph’ button to get below graph



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different color bars and in all algorithms ‘Propose DBN Algorithm’ got high performance. Now close above graph and then click on ‘Predict MEC Attack from Test Data’ button to upload test and predict attack



In above screen in square bracket can see extracted features from test data and after arrow symbol ==🡺 can see predicted value as ‘Attack’ or ‘Non-Attack’



In above screen can see extracted features contains attack and similarly by scroll down output screen we can see all test data features and predicted output