An Intelligent Data-Driven Model to Secure Intravehicle Communications Based on Machine Learning

In this paper author is describing concept to secure communication between intravehicle to avoid accidents which can trigger due to malicious intruder attack. In electric vehicles different sensors attached to different parts (steering, brakes, engine etc.) of vehicle will sense data and send to ECU (Electronic Control Units) and ECU will process data and send result back to sensors and sensors will act based on response received from ECU. Sometime some malicious users may hack sensors and ECU communication and send false information to ECU and this false information cause sensors to work improperly and can cause accidents. All existing techniques were using AES encryption to have secure packet transmission between sensors and ECU but hackers are still able to decode encrypted packets and hack communication between ECU and sensors. All ECU and Sensor communicate with each other using CONTROLLER AREA NETWORK (CAN) protocol.

To overcome from this problem author using machine learning algorithms to detect intrusion or anomaly packet received by ECU or sensors. Machine learning algorithms will be trained and model to predict attack based on request frequency received by ECU. All hackers will send packet with high frequency and priority to make ECU busy and to process high priority packets and other genuine sensors request will keep on waiting. To avoid this problem machine learning will build train model with attack class label as 1 when high frequency of packets received with same id or device. If packets receiving in normal mode then class label 0 will be assigned which indicates received packet is normal.

Author using various machine learning algorithms such as Conventional SVM algorithm, Decision Tree, KNN Algorithm and propose Social Spider Algorithm with SVM by selecting optimal features and evaluating their performance with indices such as HR (Hit Rate), MR (Miss Rate), CR (Correct Rejection Rate) and FR (False Alarm Rate).

Here HR refers to machine learning metric called TRUE POSITIVE (TP) which means classifier able to predict given record correctly as positive.

MR refers to machine learning metric called FALSE NEGATIVE (FN) which means classifier unable to predicted given record correctly

CR refers to machine learning metric called TRUE NEGATIVE (TN) which means classifier able to predict given record correctly as negative

FR refers to machine learning metric called FALSE POSITIVE (FP) which means classifier predicting negative records as positive.

For any classifier whose HR and CR is high then its performance will be consider as better and efficient. In propose work to secure CAN bus (electric vehicle communication) author is doing enhancement to SVM algorithm by analysing frequency of received packets and if received packets from same device ID has high frequency then SVM mark that records as anomaly and propose SVM performance will be evaluated using above four indices such as HR, FR, MR and CR. In propose SVM to select optimal features from dataset author is using SOCIAL SPIDER OPTIMIZATION (SSO) algorithm. In this algorithm dataset features vector will be consider as SPIDERS and fitness will be calculated between all features and features which has high similarity will be consider as related and will have high fitness score and all those high fitness score features will be selected and low fitness features will be removed out. Comparison between one features to other features will be consider as MALE and FEMALE spiders. After applying SSO algorithm we will have optimal features using which classifier can efficiently predict anomaly from new and old records.

To implement this project author using CAN BUS CONTROLLER dataset and below are the dataset examples. I saved this dataset inside ‘dataset’ folder.

**Label,Time,ID,Signal1\_of\_ID,Signal2\_of\_ID,Signal3\_of\_ID,Signal4\_of\_ID**

0,81008449.4467,id3,0.2,1.0

0,81008456.7515,id9,0.370002961208173

0,81008462.011,id5,0.17304397155148016,0.874886161240236

0,81008465.0179,id2,0.0,0.7650437326834458,0.16701107265391962

0,81008465.4139,id10,0.37071729639797957,0.0,0.8447834078508285,0.44514720267692903

1,81097085.7641,id2,0.27311659328065624,0.8500955038906693,0.1541178262275998

1,81097086.3347,id1,0.8603548622572741,0.25

In above dataset all bold font names are the dataset column names and all decimal values are the sensor values. In first column we have values 0 which indicates packet is normal and 1 means packet contains attack. Other values are the sensor signal values. We will train all classifier with above dataset. Whenever new packet received then classifier will applied on that new packet to predict whether packet is normal or attack. We don’t have any sensors to get test data so I am using below dummy values as the test data. Test data will contains only signal values and by evaluating those signal values and frequency classifier will classify/predict that test data as attack or normal. Below are some test data samples.

Time,ID,Signal1\_of\_ID,Signal2\_of\_ID,Signal3\_of\_ID,Signal4\_of\_ID

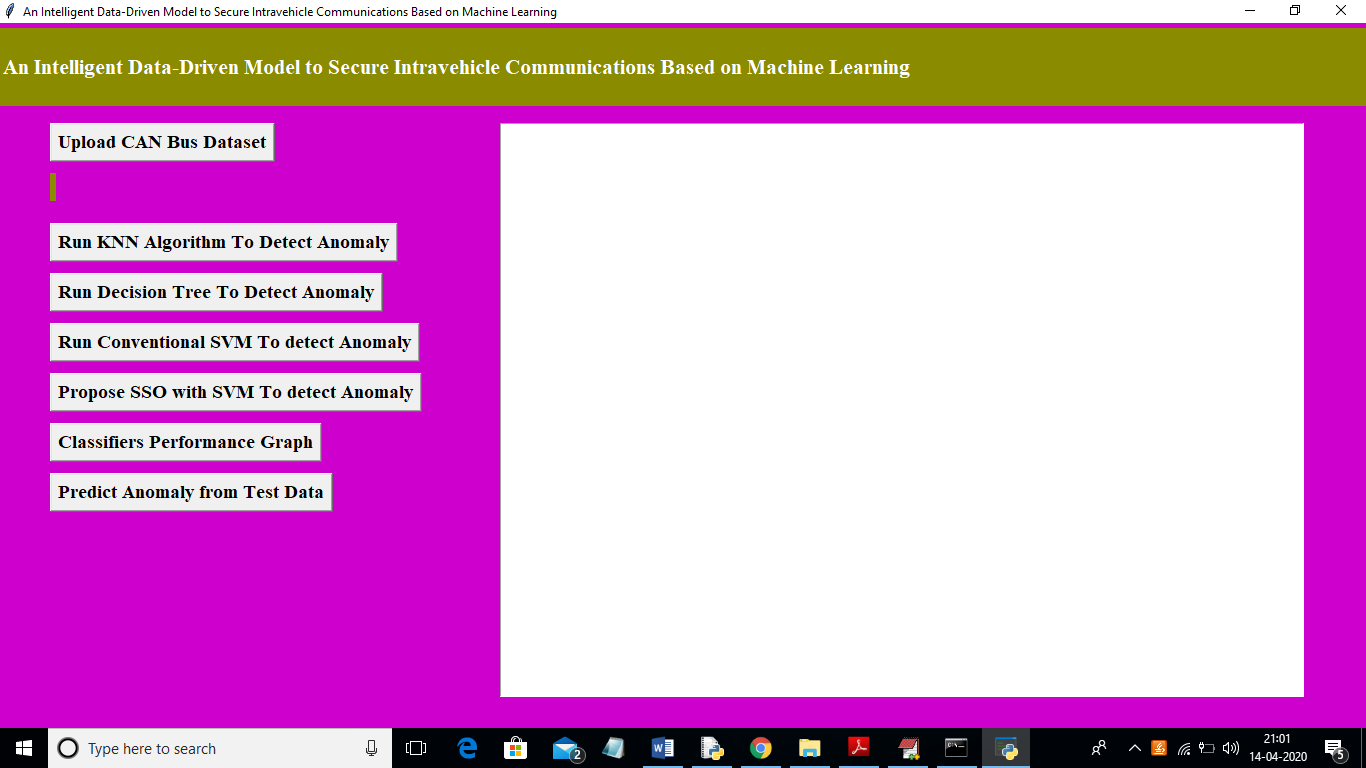
81219830.4139,id10,0.3737464000855015,0.6666666666666666,0.9991955021452382,0.4751998828306158

81219785.4139,id10,0.3724087697056715,0.4444444444444444,0.9991955021452382,0.47420213228827873

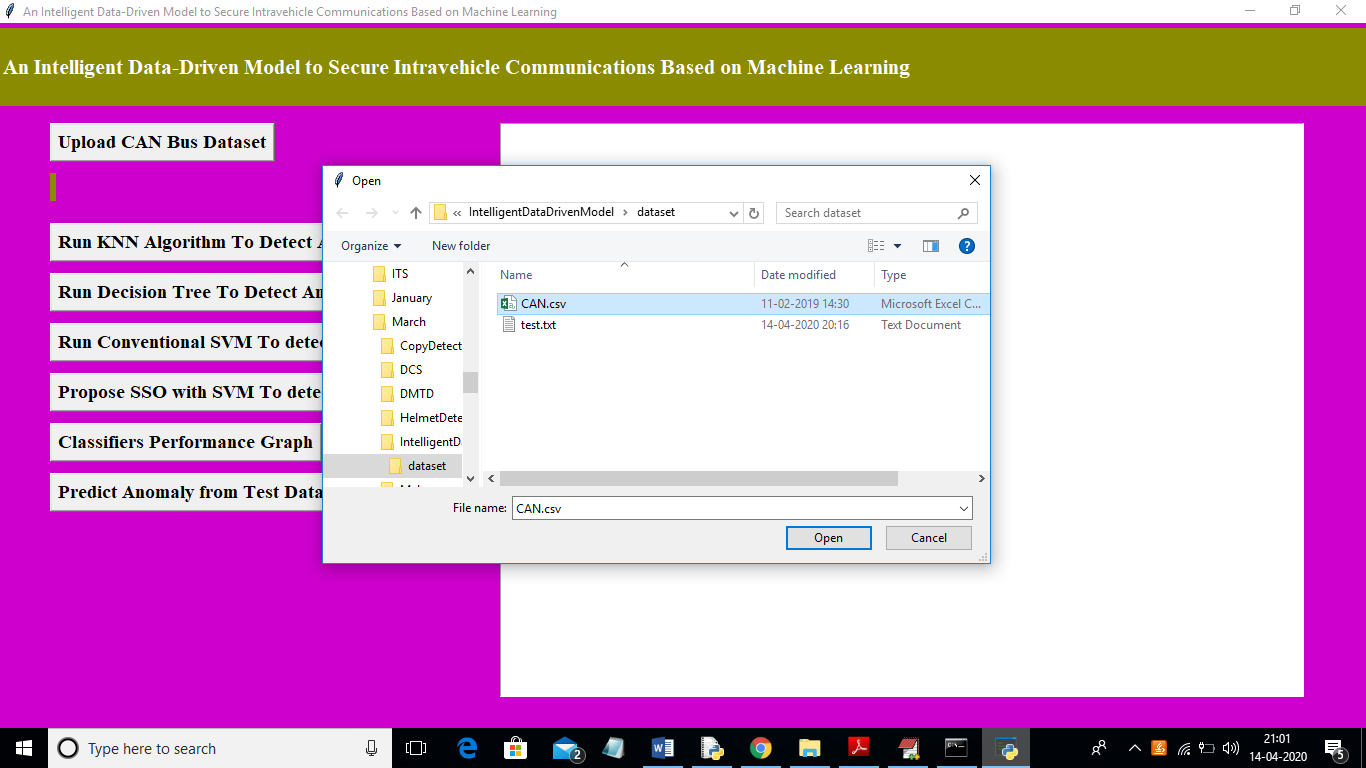
In above test data we can see there is no class label, classifier will predict its class label

Screen shots

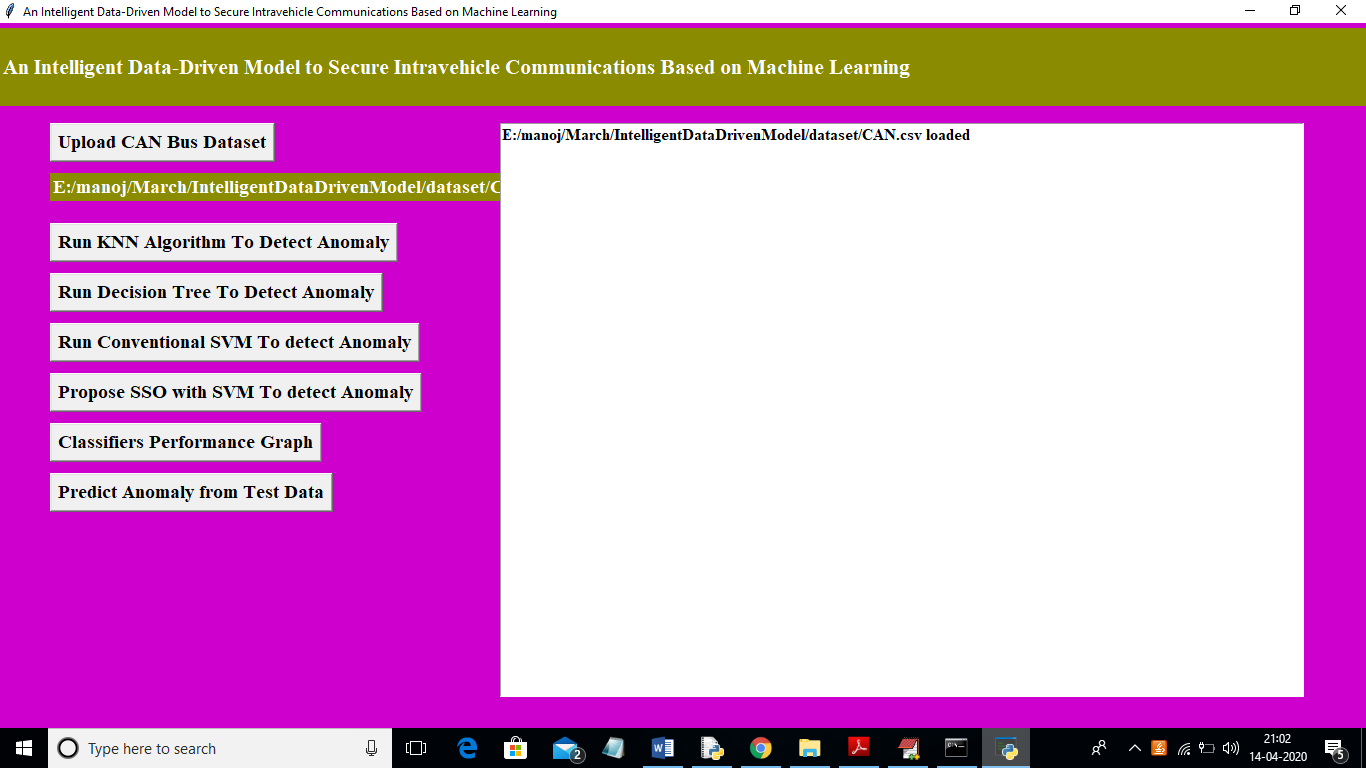
To run project double click on ‘run.bat’ file to get below screen



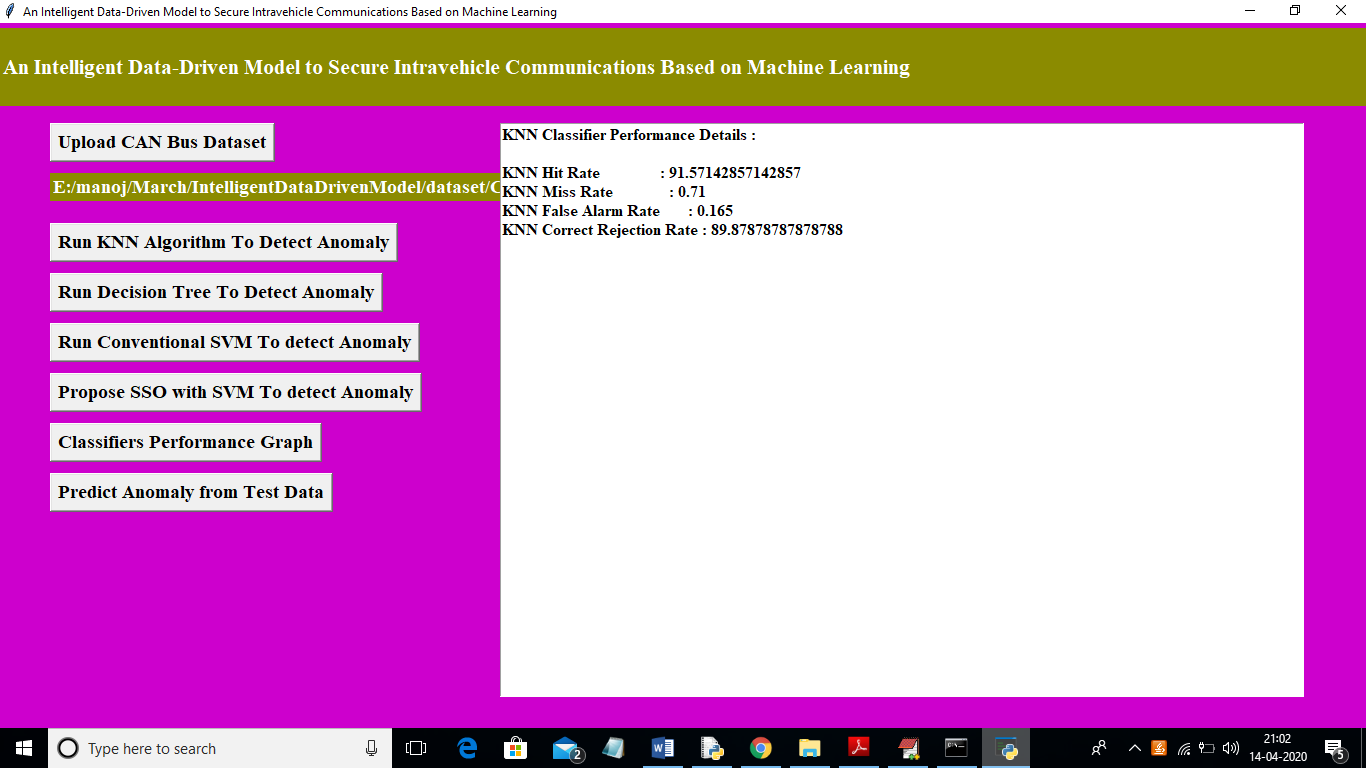
In above screen click on ‘Upload CAN Bus Dataset’ button and upload dataset



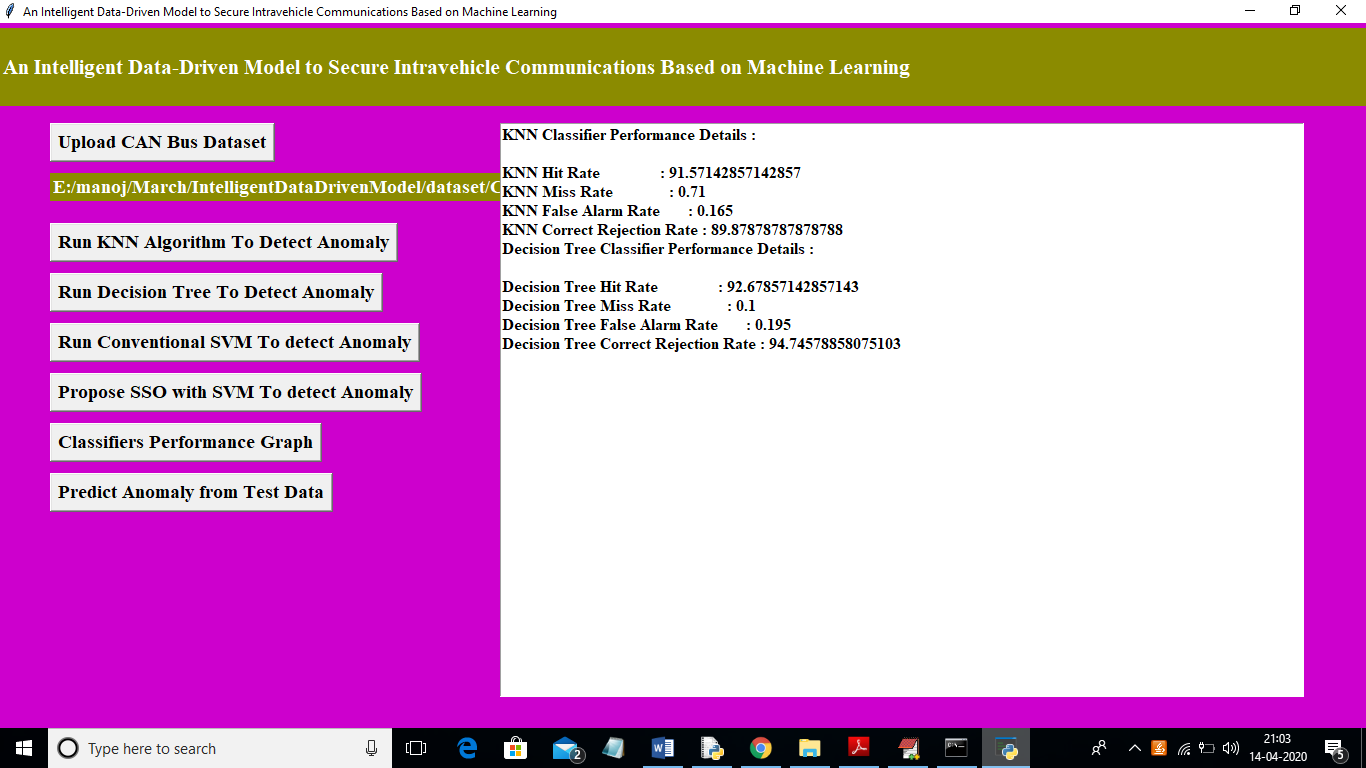
In above screen I am uploading ‘CAN.csv’ dataset and after uploading dataset will get below screen



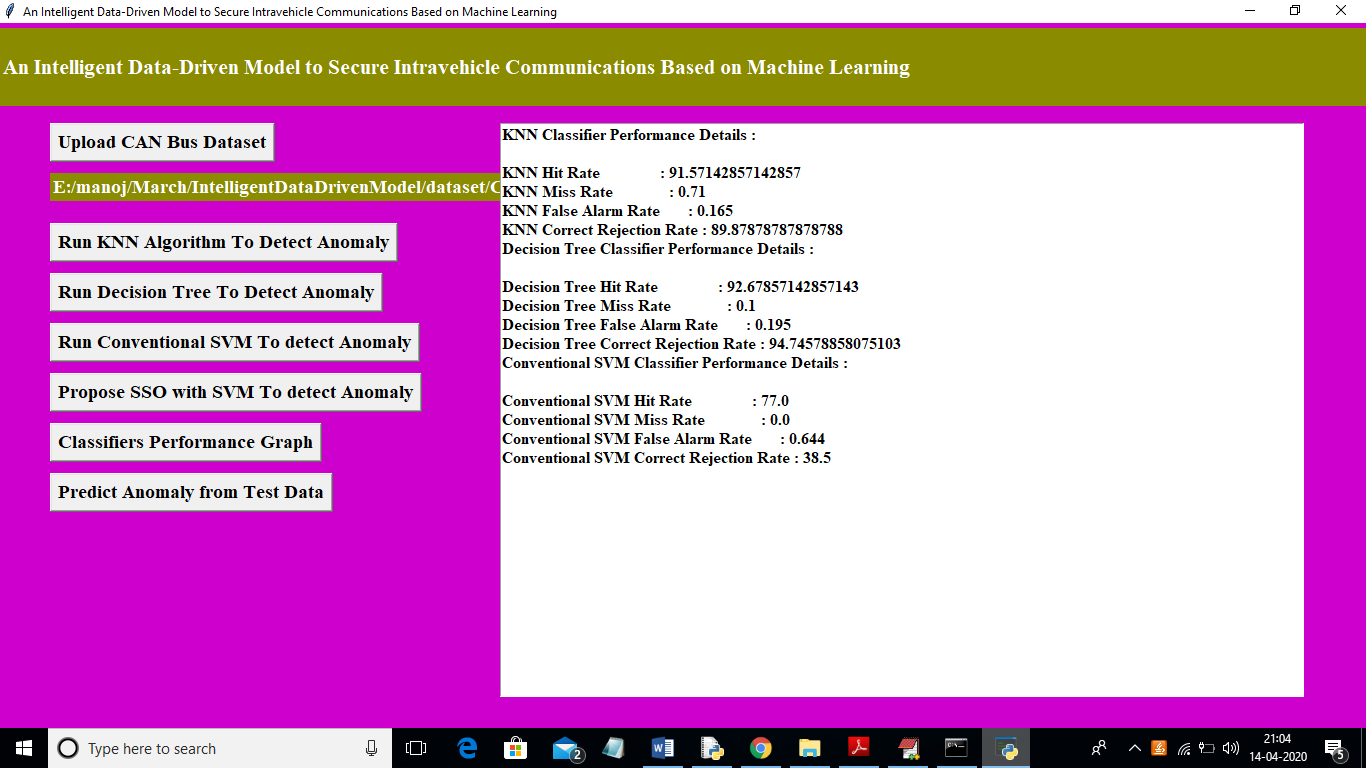
Now click on ‘Run KNN Algorithm To Detect Anomaly’ button to build KNN classifier train model to detect anomaly and evaluate its performance based on 4 indices



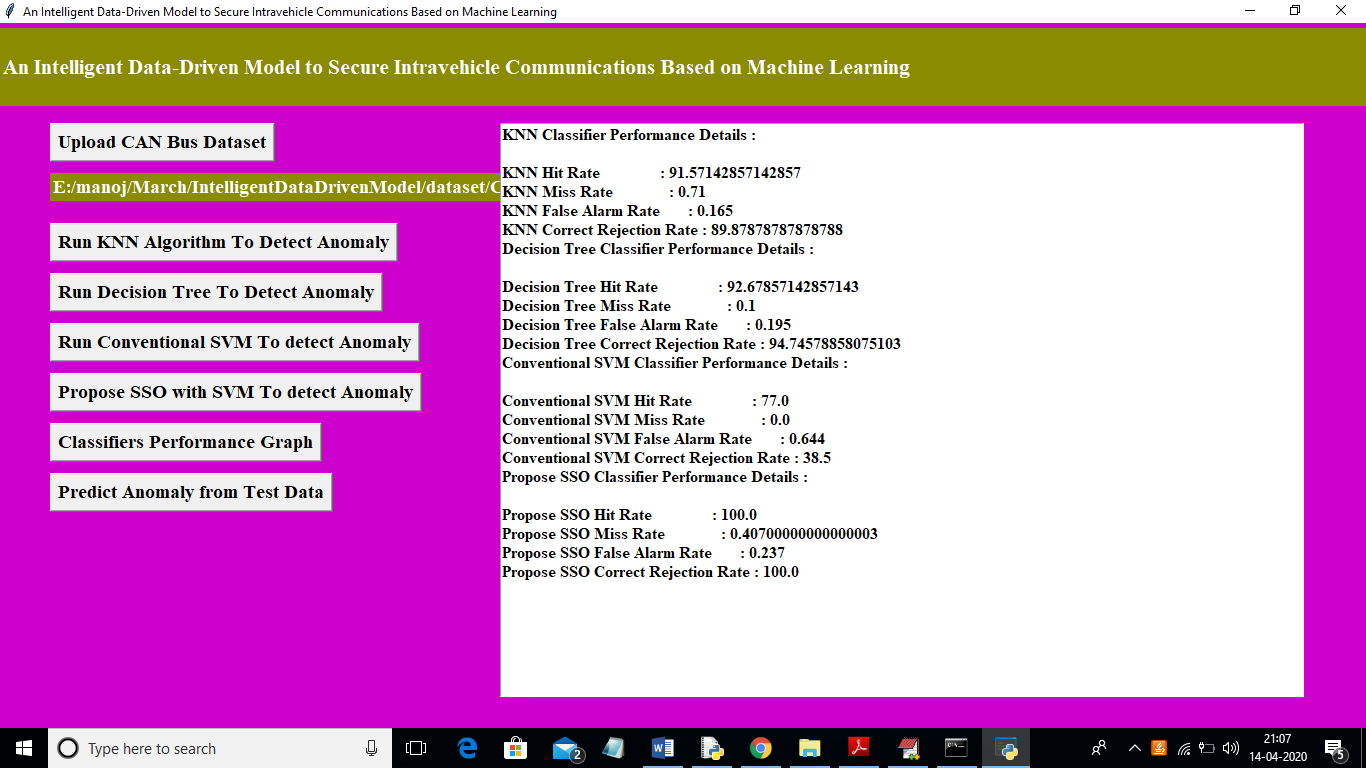
In above screen we got 4 indices values for KNN algorithm and now click on ‘Run Decision Tree To Detect Anomaly’ button to evaluate decision tree performance



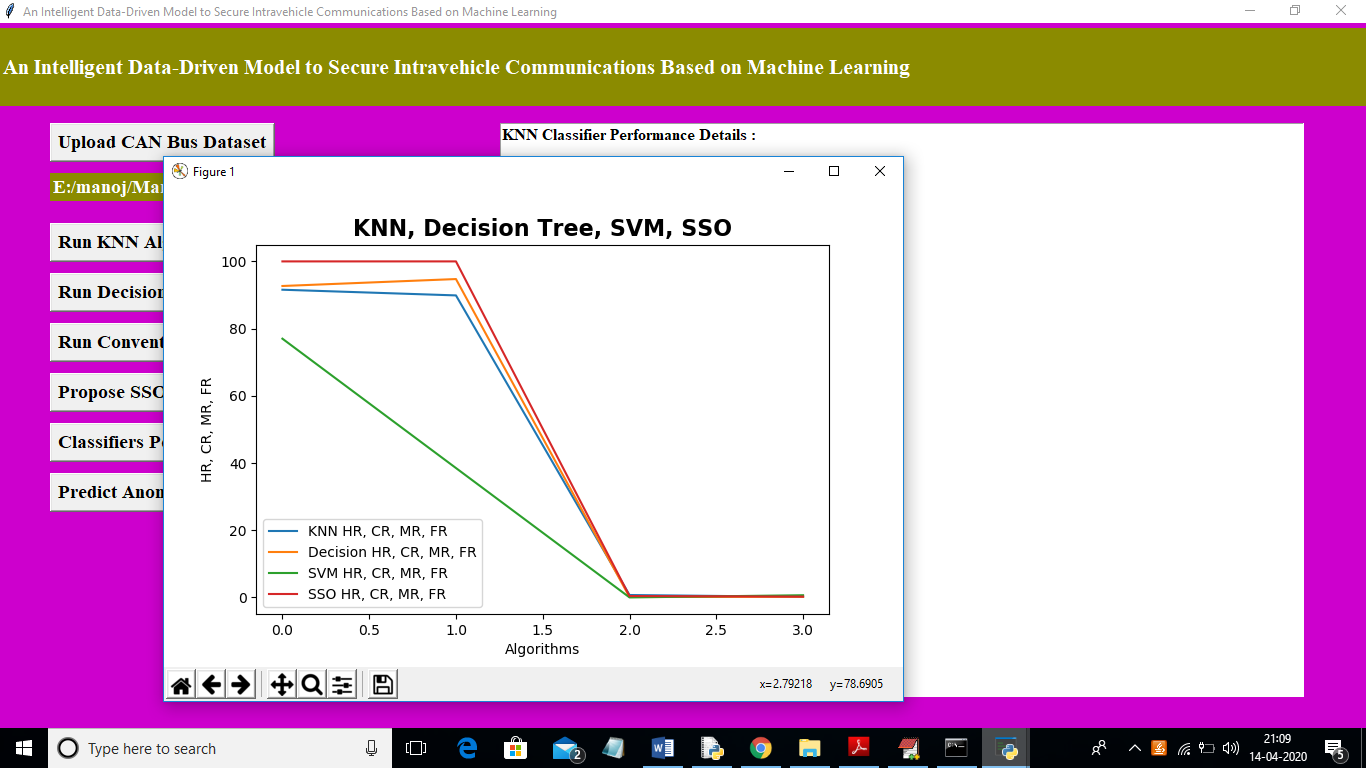
In above screen we got decision tree data and now click on ‘Run Conventional SVM To detect Anomaly’ button to evaluate conventional SVM performance.



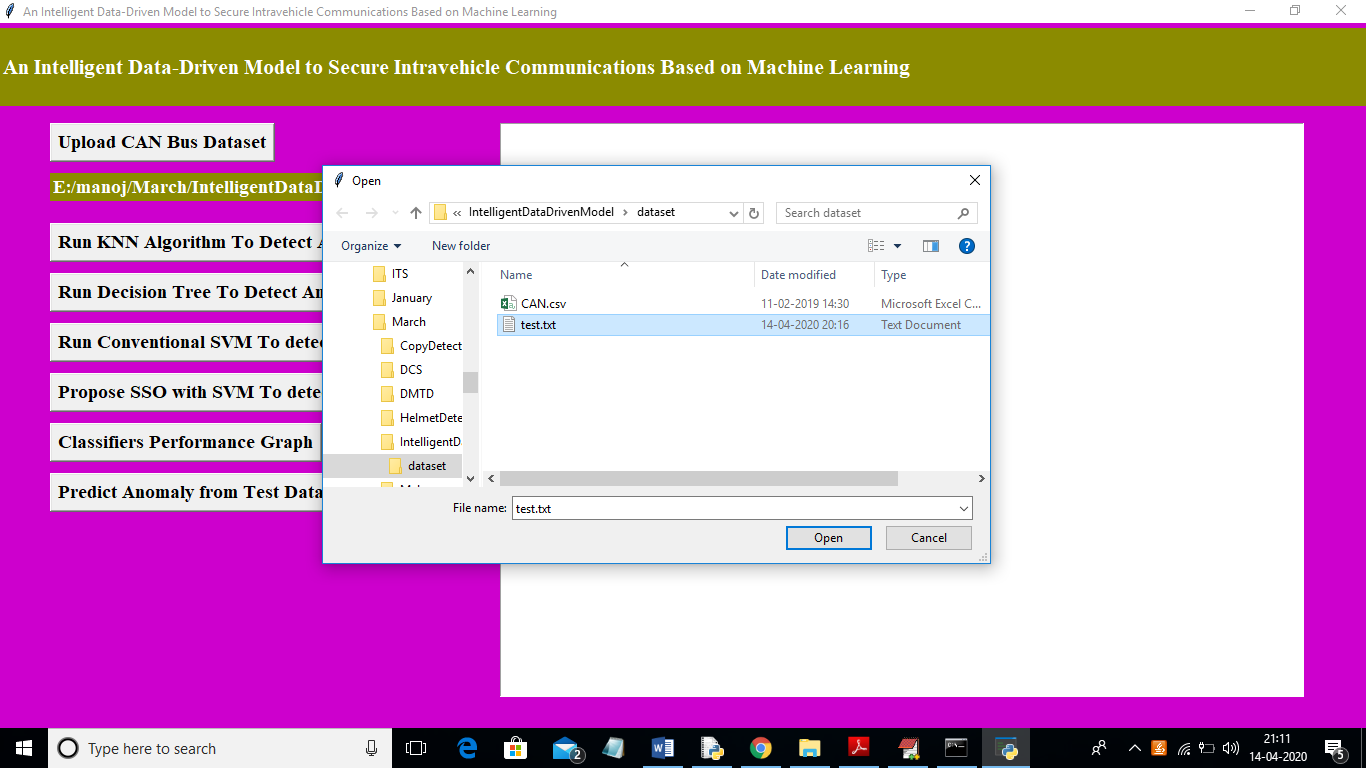
In above screen we got SVM performance data and now click on ‘Propose SSO with SVM To detect Anomaly’ button to run propose SSO with SVM classifier and evaluate its performance. (Note: when u run SSO then application will open 4 empty windows and you just close newly open empty window and keep working from first window only).



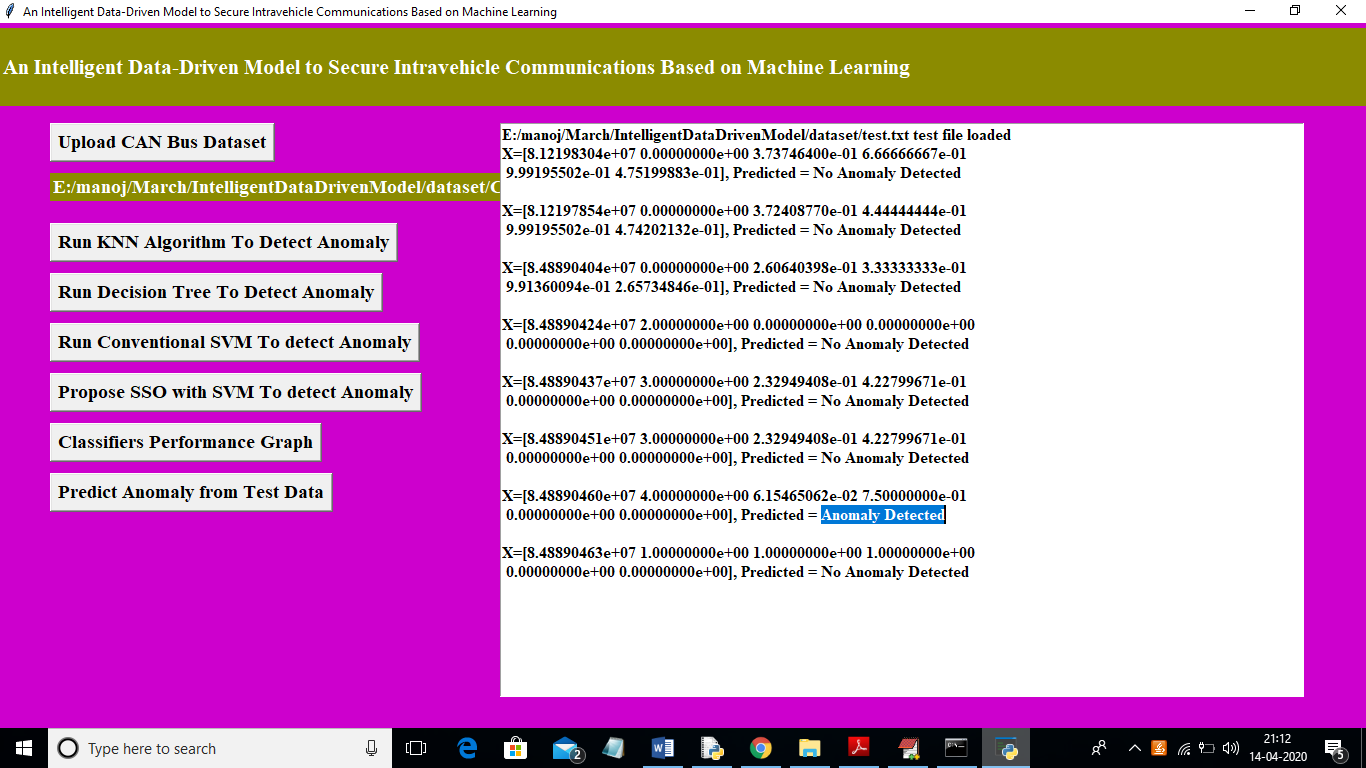
In above screen for SSO we got performance metric as 100% and MR and FR is not mandatory so we can ignore as said in paper. Now click on ‘Classifiers Performance Graph’ button to get performance graph between all classifiers



In above graph propose SSO has given high performance compare to other algorithms. In above graph y-axis represents HR, MR, FR and CR values. Now click on ‘Predict Anomaly from Test Data’ button to upload test data and predict it label



In above screen I am uploading ‘test.txt’ file and now click on ‘Open’ button to predict uploaded test file class label.



In above screen in text area we can see uploaded test data and its predicted class label. All records contains normal packet data accept one record. So by using machine learning algorithms we can analyse packets and if packet contains attack then we ignore processing such packets.