Anomaly Detection in CAN-Bus Data

Control Area Network(CAN)-Bus is a vehicle bus to allow micro-controllers, ECUs and various other devices to communicate directly with each other. The CANBus accepts data in form of **independent packets** which can store data of 8 Bytes and an ID field to represent the ECU from which it is coming. It is a **broadcast bus**.

All the modern cars are making all the critical controls such as Brakes, Engine, Accelerator and many more connected to the CANBus because it provides a very **fast medium** to communicate with each other within the car. This speed in communication comes at the price of security. There is **no security** whatsoever for the bus except a checksum which can easily be tampered with. So we tried to develop methods by which the ECUs can protect itself (from any malicious data received by it) by analysing the previous history data.

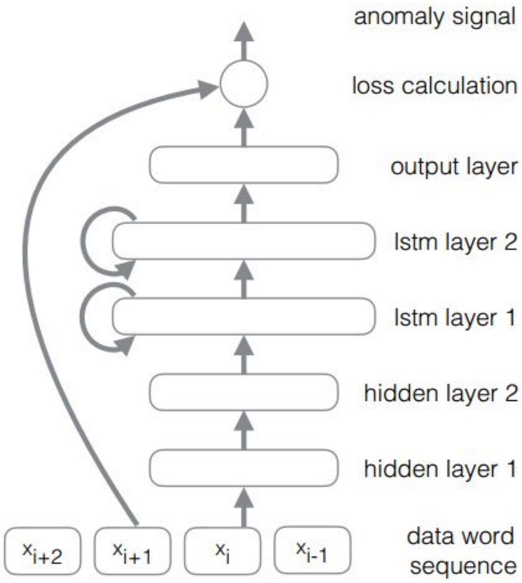
We had purchased CAN Bus shield and Arduino Uno to **emulate an ECU**. We used the OBD port 2 to DB9 connector to connect CAN Bus shield to a car(Honda City) to sniff the CAN data packets. We were able to successfully sniff CAN packets from 38 different ECUs.

It was observed that the packets in CAN Bus that come from some particular ECUs have **sequential relationship** between them. That is, if one ECU sends some data then it triggers some other ECU to send some other data either in response to the data from the former ECU or to just become synchronized with that ECU. To exploit this property we used we applied various algorithms to find patterns within the different combinations of data.

We applied Data Mining, Machine Learning and Deep Learning Algorithms in Python on these data to get any pattern matching for valid and malicious data.

We implemented **Deep Neural Networks(DNN)** to find patterns within the sequence of data streams and experimented with changes in layers of neurons and its controlling parameters. It gave us satisfactory results.

Then we used **Long-Short Term Memory(LSTM)** neural network to retrieve patterns. After analysing the data, we came up with the architecture model shown below.
We applied LSTM with binary as well as with categorical cross entropy loss function on different sets of data. According to the type of LSTM used, the the input data and the neurons to be used in each layer of the model were modulated. This model gave us the most promising results for all the major types of anomalies against which we tested the model.



LSTM based anomaly detector model

On similar lines we experimented with **Gated Recurrent Unit(GRU)**, but the results of it were not on par with the results of LSTM.

Then we also experimented applying **Gaussian Mixture Model(GMM)** clustering and **One Class Support Vector Machine(one class SVM)** classification but, as much of the CANBus data also contains continuous valued data which is mapped to different other bytes. Hence, the randomness in the data is increased. For this reason, the classification between the real and malicious data was not significant using these two models. Only after analysing its results we were able to conclude the reason for its poor accuracy. And it was for the same reason as above, that we had not used decision trees or random forests to recognize patterns for our data.

We also used data mining technique of **Apriori algorithm** to find packets that occur in groups and if it was found that the received data doesn't have a minimum required confidence, we marked them as malicious. Entropy analysis further helped us to easily classify parts of data which was majorly consistent throughout its existence.

We used keras API with theano for LSTM and GRU implementations in python. For DNN, GMM and one-class SVM the sklearn library in python was very helpful.

I hope this is able to give an overview of my final year project.