ECE-5831 Final Project: Classifying CAN Signal to Predict ECU Input

Jonathen Robey
CECS Department
University of Michigan
Dearborn, USA
jdrobey@umich.edu

***Abstract*—The final project for class ECE – 5831 is a machine learning project. This project aims to demonstrate the student's ability to complete a ML problem from start to finish. This includes preprocessing data, research, choosing a ML model, choosing features, programming and testing the chosen model and reporting the accuracy of the model. The data set for this project comes from eight (8) ECU's. The goal of the project is to be able to classify these ECU's based on the CAN signal produced.**

***Index Terms*—Electronic control unit (ECU), controller area network (CAN), machine learning (ML), mean squared error (MSE), Gaussian distributed analysis (GDA).**

I. Introduction

Modern vehicles are increasing in the number of mechanical parts that are controlled by ECU's. These ECU's communicate with CAN signals. These signals do not describe which ECU is communicating. Because of this, a remote hijack of a vehicles ECU could result in control of the vehicles mechanical parts. For example, the ECU near the console used to control climate, navigation, music, etc. could be hijacked and used to control the fuel or power to the engine. The receiver at the engine gets the signal but cannot determine if the signal was sent by the ECU at the pedal or the ECU near the console. This example calls for action. To prevent hijacking, it is imperative that the CAN signal from each ECU be classified and used to determine which ECU is sending the signal.

The data for these CAN signals came from Dr. Azeem Hafeez of the University of Michigan - Dearborn. Upon plotting the data, it is apparent that the CAN signal for each ECU is that of a rectangular wave function as shown in $Fig.$ I. Overlapping each data set on the same plot showed that this data will need to be preprocessed.

This project outlines two methods of achieving ECU classification through ML. The first method is based on GDA while the second method is based on a regression model. In order to properly train and test these models the data set used for each method will need to be randomly split into a training set and testing set. The accuracy of these models will be the ratio of outputs guessed correctly to the total number of outputs in the test set. The accuracy of a model will be validated by running the model multiple times with new training/test sets through randomization or increasing the ratio of data held for training to the data held for testing.

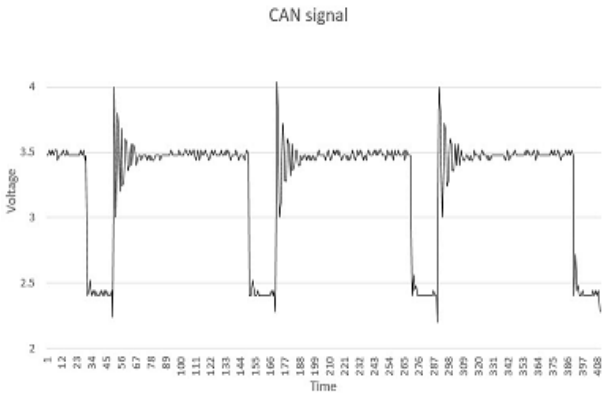

Fig. 1.

II. Data Processing

As mentioned in the introduction, each collected sample of CAN signal needs to be processed. To do this, each signal was imported into a Python script. The data was then stored in an array. Starting at the first index of the array, the value is checked to see if it is less than three. This was used to indicate whether the signal recording was initiated during a dominant voltage (3.5) or a recessive voltage (2.5). If the starting voltage was dominant then the array was trimmed down until the recessive signal began. If the starting voltage was recessive then the array was trimmed past the next section of dominant voltage until the recessive signal began. This made it so that each signal began with the recessive voltage. This was done for all eight ECU's where each ECU had it's CAN signal recorded 30 times resulting in 240 CAN signals. After processing the data, each CAN signal recorded was reduced to an array of 480 elements starting with the recessive voltage. The last column

of the final data set is categorical and labels the data from 1-8 depending on the ECU that the CAN signal belongs to.

III. Method 1: GDA

The first model for predicting source ECU used GDA. Gaussian distribution or normal distribution will create a bell curve centered at the mean or average of a data set. For this analysis, each time iteration for each ECU was the data set.Therefore, each data set will have 30 values corresponding to the 30 CAN signal records for each ECU. Before calculating the mean and standard deviation of each data set the raw data was randomly spit into a training and test set. The training set consisted of 70 percent of the raw data while the test set consisted of 30 percent of the raw data. Next, the mean and standard deviation of each data set was calculated. The result for each calculation was a list of eight arrays. Each array had 479 elements corresponding to the time iterations of the raw data set. These were the mean and standard deviation values of each time iteration for each ECU. This data is shown in $Fig.$ 2. From this data the GDA was performed on the test set. The test set in question needed to be compared to the mean and standard deviation set of each ECU. To do this, the values of the test set were subtracted from mean values of the ECU being compared too. Next, the absolute value was taken of this difference. This difference shows how far the test set is away from the mean of the ECU in question. From this, we can use the standard deviation set of the same ECU to predict the likelihood that the test set belongs to the a particular ECU. This was done by recording the number of differences that were outside of 3.2 standard deviations from the mean set of the ECU being compared. This was a simplified way of determining the likelihood that a time iteration belonged to the ECU in question. This comparison was done for each ECU. Once the test set was compared to all eight ECU's the result was an array of eight elements consisting of the sum of differences that were outside of 3.2 standard deviations from the mean of the ECU being compared. The element with the lowest value was selected as the predicted ECU. This means that the training set had a higher probability of being that ECU.

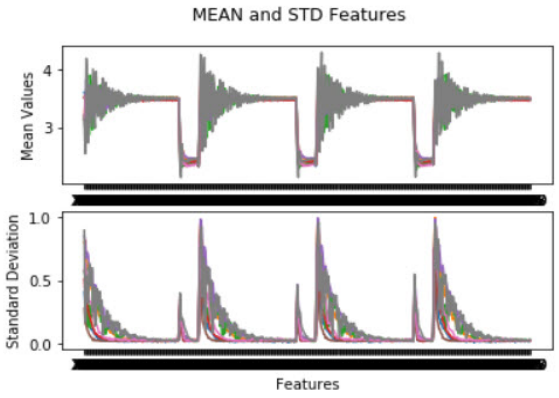

Fig. 2. $Mean\ and\ STD$

IV. Method 2: Regression Model

The second method for predicting source ECU uses a regression analysis to predict the function that best fits the data set for each ECU. The test set will be compared to the predicted function of each ECU and the the function resulting in the least error will be the predicted ECU. For this model, it is essential that the data set be limited to a single wave of dominant voltage (3.5). This data trim was done by inspection, but could be done through an algorithm of logical conditions.Next, the data will need to be normalized. To do this, each datum point was subtracted from 3.48 (value chosen from inspection of data) and have the absolute value taken. When this is done, the data of each CAN signal will consist of dominant voltage data normalized around zero. This is depicted in $Fig.$ 3.

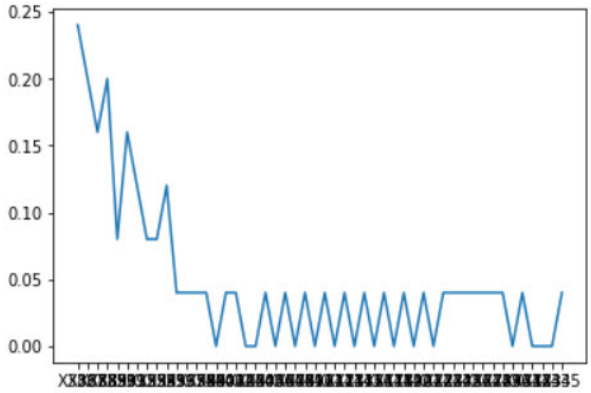

Fig. 3.

From this plot, it was determined that an exponential decay function would be the best fit for regression analysis. The following equation was used in the analysis:

$$y_{(i)}(t) = a_{(i)} * e^{(k_{(i)} * t)}. \tag{1}$$

Here, the subscript i represents the ECU number, a represents the starting value at time equal to zero, e is Euler's number, and k is the decay rate.

Now, a regression learning model with the decay function above can be created. For this model, the a value was calculated by taking the average value at the first time iteration for each ECU. Next, the decay rate was was initiated as -0.1. This was chosen by inspection. After these two weights of the function have been determined they can be plugged into the equation above and used as the first prediction for the first ECU. Then, all training CAN signals for the first ECU is subtracted from this function. This is the error. The error at each time iteration is summed up resulting in a single value. If this value is negative then the decay rate isn't large enough and the k value will be adjusted by $mu = 0.0001$. If this value is positive then the decay rate is too large and the k value will be adjusted by $m * (-1)$. This loop will be repeated until the number of epoch iterations is complete. For this model, the number of epochs was chosen to be 5000. After running this

model for each ECU *Fig.* 4 shows resulting fit functions for each ECU.

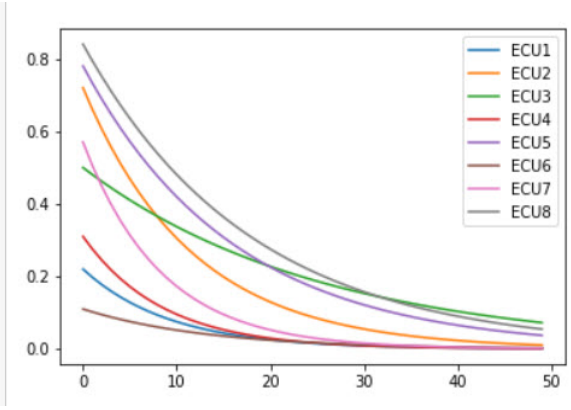

Fig. 4. *ECU Prediction Functions*

From here, the classification is simple. Take the test set and subtract it from an ECU fit function and square each value. This is the squared error. Next, take the average of all the squared errors to obtain the mean squared error or MSE. The ECU fit function that results in the lowest MSE is the ECU chosen for classification.

V. Results: Method 1

The model for method 1 using GDA was computationally cheap. The time it took to run the model was less than three seconds. The accuracy of the model, though, was less fortunate. After running the model multiple times with different training/test sets, the average accuracy was about 75 percent. The confusion matrix for one of these experiments is shown in *Fig.* 5.

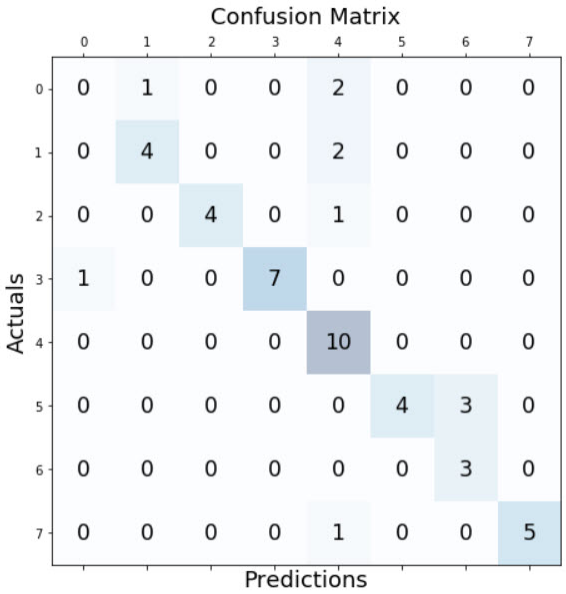

Fig. 5. *Confusion Matrix Method* 1

From the confusion matrix, it is apparent that ECU 1 and 6 where predicted incorrectly the most. Also, most of the incorrect predictions were classified as ECU 5.

VI. Results: Method 2

The results of method 2 where more encouraging but came with a cost. The average computational time for this model was 18-20 minutes. This is a significant amount of time when compared to the three seconds it took to complete the model in method 1. The computational expense is worth it because this model resulted in an average accuracy of 98 percent. This model was ran on multiple different training/test sets. The confusion matrix for one of these experiments is shown in *Fig.* 6

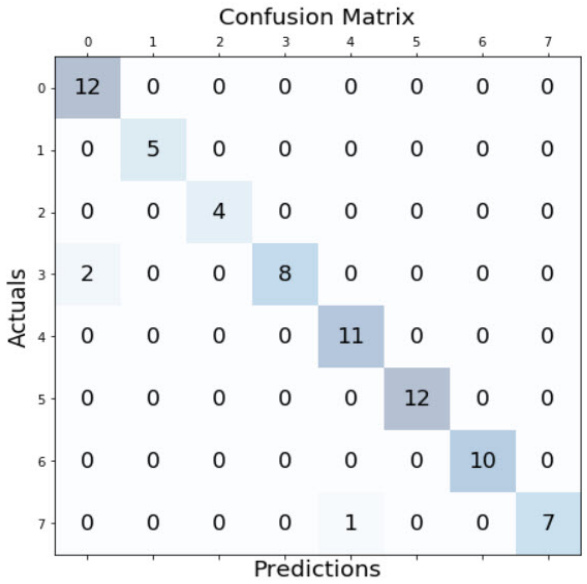

Fig. 6. *Confusion Matrix Method* 2

Looking at the confusion matrix, there were only two ECU's that were miss classified; ECU 4 as ECU 1 and 8 as ECU 5. Comparing this result with the ECU fit functions in *Fig.* 4, the results make sense if the data from those particular training sets where skewed to the left or missing the first few datum points that corresponds to the correct time iteration.

VII. Conclusion

Two ML methods were used to identify source ECU from CAN signal. The first of these methods used a model structured around GDA. The second method used a model structured around regression analysis. The first method resulted in adequate prediction accuracy but was computationally quick. The second method had outstanding accuracy but was computationally expensive. The first method was greatly influenced by the training/test set given the accuracy of each experiment varied greatly from 60 percent up to 90 percent. This suggests that the accuracy of this model has potential to increase and become more consistent as more CAN signals are recorded and added to the raw data set. The additional data should not pose a

threat to the computational cost. When observing the confusion matrix of the second method and comparing it to the ECU fit functions, it seems like the accuracy could be improved if the pre-processed data was done correctly. This could be solved by updating the algorithm that sorts and processes the data. Alternatively, the a value in equation 1 could be added as a randomized weight. This would add to the complexity of the model, but it would increase decrease the error of the functions by giving it more flexibility. In addition to that, the decay rate could be used as another feature to check if the guess ECU has a decay rate closer to another class. This may involve the addition of probability theory.

Overall, the results of each method were satisfying and the information gathered from each method have shown great insight into the possibilities of accurately classifying ECU's. This is a small but necessary step towards the protection of in vehicle controls from criminals.

VIII. Acknowledgements

I would like to thank my professor Dr. Azeem Hafeez for the excellent teaching of this course. All knowledge in this paper was obtained from his clear and informational lectures. Any questions regarding in vehicle ECU's, CAN signals, or how the data used in this project was obtained should be directed to Dr. Azeem Hafeez.