ESTIMATING INSTANT FUEL CONSUMPTION BY MACHINE LEARNING AND IMPROVING FUEL CONSUMPTION

(MAKİNE ÖĞRENMESİ İLE ANLIK YAKIT TÜKETİMİNİN TAHMİNLENEREK YAKIT TÜKETİMİNİN İYİLEŞTİRİLMESİ)

by

Buğra ŞEN, B.S.

Thesis

Submitted in Partial Fulfilment

of the Requirements

for the Degree of

MASTER OF SCIENCE

in

COMPUTER ENGINEERING

in the

GRADUATE SCHOOL OF SCIENCE AND ENGINEERING

of

GALATASARAY UNIVERSITY

Jun 2020

Table of Contents

List of Figures

List of Tables

ABSTRACT

With the development of the internal combustion engine, our lives have changed significantly. Although their use has facilitated the development of humanity they have become one of the major contributors to environmental pollution and resource utilization. Although electrical vehicles seem to be the future of transportation, internal combustion engines will continue to be a part of our lives for the foreseeable future. There are a vast multitude of techniques to enhance a vehicles performance and economy, from engine modifications to aerodynamic changes. However, it is difficult to verify the changes without knowing the fuel consumption of the vehicle under various situations. Modern cars are very technologically advanced and rely on sensors and actuators which communicate with control units, therefore it becomes possible to obtain data by using the vehicle sensor data from the controller area network (CAN) bus. Due to its bus structure, it is possible to reach real-time detailed data from sensors inside the vehicle such as O2 sensor voltage, fuel pressure, catalyst temperature etc. This study aims to predict the instantaneous fuel consumption by collecting a large-scale vehicle sensors' data and create a model with machine learning algorithms with the goal of better understand how the multiple variables influence the instantaneous fuel consumption. With this predictive model, it is possible to make fuel experiments and receive improvement results easily such as inlet water injection, side mirror aerodynamic optimization etc. This approach and the experiments can also support original equipment manufacturers in developing and marketing this technology in the future. This work may lead the way to a cleaner environment due to more economical and less polluting vehicles.

ÖZET

İçten yanmalı motorların gelişimi hayatımızı önemli ölçüde değiştirmiştir. Her ne kadar kullanımları insanlığın ilerlemesini kolaylaştırmış olsa da, çevre kirliliğinin ve kaynak kullanımının en büyük sebeplerinden biri haline gelmişlerdir. Elektrikli araçlar ulaşımın geleceği gibi görünse de, içten yanmalı motorlar öngörülebilir gelecek için hayatımızın bir parçası olmaya devam edecek. Günümüzde motor modifikasyonlarından aerodinamik değişikliklere kadar araç performansını arttırmanın ve yakıt tüketimini azaltmanın bir çok yolu mevcutur, ancak çeşitli durumlar altında aracın ne kadar yakıt tüketimi yapıp yapmayacağını doğrulamak zordur. Günümüz araçları bir çok sensör ile donatılmış olup bu sensörlerin birbiriyle haberleşebildiği kontrol ünitelerine sahiptir. Bu sebeple merkezi haberleşme veriyolundaki (Can) araç sensörü verilerine ulaşmak mümkün hale gelmektedir. Bu protokol sayesinde, aracın içinde bulunan sensörler vasıtası ile hava yakıt karışımı oranı, yakıt basıncı, katalizatör sıcaklığı gibi ayrıntılı verilere ulaşılabilmektedir. Bu çalışma, makine öğrenme algoritmaları ile büyük ölçekli araç sensör verilerinin toplanması sonrasında anlık yakıt tüketimini tahmin etmeyi amaçlamaktadır. Oluşturulan model sayesinde sensörlerden elde edilen değişkenlerin anlık yakıt tüketimini nasıl etkilediği daha iyi anlaşılacak ve su enjeksiyonu, yan aynaların sökülmesi gibi yakıt tüketimi azaltması beklenen yöntemleri daha az efor harcayarak deneyimlemek mümkün hale gelecektir. Bu yaklaşım gelecekte araç üreticilerinin yakıt tüketimini azaltma amaçlı yapmış oldukları çalışmaları destekleyerek markete yeni yeknolojiler kazandırabilir. Bu sayede daha ekonomik ve daha az karbon salınımı yapılarak daha temiz bir çevreye sahip olabiliriz.

1 INTRODUCTION

The carbon emissions from the transport sector account for a large amount of total carbon emissions in the world. In the study of (Fuglestvedt et al., 2008) , they find that, since pre-industrial times, transport has contributed 15% and 31% of total man-made CO2 and O3 forcing, respectively. The current emissions from transport are responsible for 16% of the integrated net forcing over the next 100 years for all current man-made emissions. Various methods are being tried to reduce this situation which has an effect on fuel consumption. Examples of such methods are water injection and aerodynamics improvement etc. Such practices can contribute to fuel economy by reducing carbon emissions as well as consuming less fuel. However, comparing different methods for fuel consumption requires great effort. A lot of tests are usually required to measure how much the applied method benefits compared to the old one. In this context, it is necessary to better understand the vehicle fuel consumption in various conditions. Currently most manufacturers utilize bench testing of the engine to vary its parameters, or wind tunnels to measure the aerodynamics of vehicles. Both these methods are costly and may not always correlate to the real environment in which the vehicle is being operated. The general problem description is to examine a large number of attributes describing fuel consumption situation and to deploy machine learning methods to find a relation between all the factors and fuel consumption. It is aimed to understand the results of the applied methods with minimum effort and thanks to this research, we have built a framework that can identify factors that may influence fuel consumption in a way that solves the question of effort expenditure.

Nowadays, modern cars contain complex electronic and mechatronic systems that monitor and control various aspects of the vehicle. There are different modules in the cars that work together to increase the safety and comfort of the driver. These different modules communicate over one or multiple Can (Controller Area Network) Bus which is a central network system in automobiles and other technological vehicles, connecting all modules working in the vehicle to the system and thus the components in the vehicle work together efficiently and effectively.(Szalay et al., 2015) Thanks to the Can Bus protocol, it is possible to understand what is happening in the engine at any moment using this protocol, it will be possible to develop a system that will log the data produced by the sensors and then implement the methods of machine learning using this data. At this point, we are faced with a problem. Each manufacturer keeps information about the vehicle differently, and it can often be impossible to find out how this information is kept. We used machine learning algorithms and reverse engineering methods to solve this problem and make sense of the Can Bus data.

In order to verify the study, we determined a route suitable for distinguishing different driving characteristics such as descending and climbing hills, accelerating and going with a constant speed, where we performed various measurements.

In this study, we have developed a module that can record can bus data, make sense of the data by using machine learning approaches and reverse engineering approaches, and automatically and accurately identify the factors that may affect fuel consumption.

In our work we provide solutions to the following specific problems:

1. Can Bus data varies from manufacturer to manufacturer. The information about which ID contains which data needs to be resolved.

- Fuel consumption can be estimated without Can Bus data being resolved but since our aim is to reduce the fuel consumption of the vehicle, the factors which can influence this need to be understandable and measurable. We solved this problem by using reverse engineering and machine learning approaches together, and we propose a method for identifying the can ID's, and detecting those which have a big impact on fuel consumption.
- The analysis of important ids of the 2012 Chevrolet Aveo vehicle will also be a valuable resource for similar studies that can be done. In this way, the effort necessary for similar studies has been further reduced.

2. Measuring the fuel consumption of a vehicle is a very time consuming and costly process. This process requires laboratory testing of the engine and expensive

wind tunnel experiments. Fuel consumption data obtained from a track does not always yield the same results as environmental conditions change. Furthermore the generated fuel consumption values may not meet the actual use case of the vehicle.

- In order to solve this problem and to predict fuel consumption in a practical way, we have build a forecast model that can solve this problem by using machine learning approaches that take Can Bus data as input. Thanks to this model, regardless of external factors, we can estimate instant consumption in real time with 94% R squared performance metric regardless of vehicle operator.
- Thanks to the model that we destined the input parameters of the model can be manipulated to determine the effect of various environmental or vehicular conditions such as engine temperature, intake air temperature, boost pressure etc..

3. It is complex and costly to find improvements to vehicles that reduce fuel consumption. For instance, it requires a certain amount of effort to fully understand the impact of these changes. The ability to determine whether the improvements to be made are worth expenditure on research and development will offer tremendous returns.

- Our model is also capable of highlighting the most significant factors in fuel consumption, therefore facilitating the effort/benefit calculations of potential changes to vehicles.

2 LITERATURE REVIEW

In this section, previous fuel consumption estimation studies and applied methods are grouped and data sources with methods are detailed in sub-headings. We have explained the related studies under 3 headings: simulation based studies, statistical methods and machine learning approaches. Each group has its pros and cons, It is not only possible to establish a successful fuel consumption model with statistical methods, but also to implement machine learning approaches without performing some statistical analysis. In this study, it was tried to follow a path that combines all these approaches.

2.1 Simulation based approaches

Much of the previous work in fuel consumption prediction consists of simulation based approaches with heavy trucks (Sandberg, 2001). They use a simulation program called the Scania Truck and Road Simulation (stars) for the analysis of fuel consumption in heavy vehicles. This simulation system requires vehicle and driver specic conguration of the model to be able to perform prediction fuel consumption. It takes many hours of highway driving and should be used for statistics, comparisons of fuel consumption, emissions and average speeds. To facilitate simulations of long driving distances with short calculation times the vehicle model needs to be rather computationally efficient. Simulation based approaches have the problem that it is difficult to measure environmental and mechanical effects. It means that to handle whole production range of engines, gearboxes, central gears, tire treads and dimensions are a costly and difficult to measure. Another problem that should be faced is that they take a long time to run and require considerable manual conguration in order to perform prediction. Simulation based applications could made to work on roads which have previously been measured many times. The approaches of Nasser Weibermel were to define mathematical formulas for the fuel

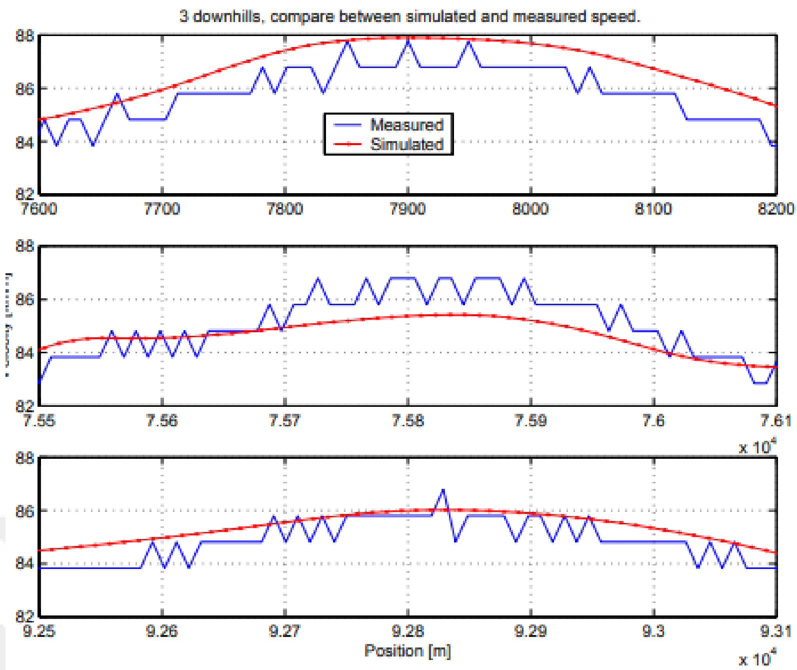

Figure 2.1: Comparsions between measured and simulated speed in three different down hill slopes.

consumption with modifying a simulation based models but this approach also increase the prediction complexity and potentially make it signicantly slower (Nasser et al., 1998). They have achieved % 2 error rate in the pre-measured highway driving in their study.

2.2 Statistical based approaches

In general, research on statistical analysis of fuel consumption in literature can be classied according various perspectives. The purpose of the research area has been decisive, e.g. driver recognition, maneuver recognition, aggressive or eco-friendly driving detection, etc. For the analysis, usually GPS locations and Can Bus data are used.

In the paper of (Fugiglando et al., 2019), they collect a dataset which has 2135 hours of driving data for each of the 2418 sensors. The data collection phase took place in 2014 with a total of 55 days of experiment. Cars were picked up by the drivers in a central deposit and had to be returned within the same day.

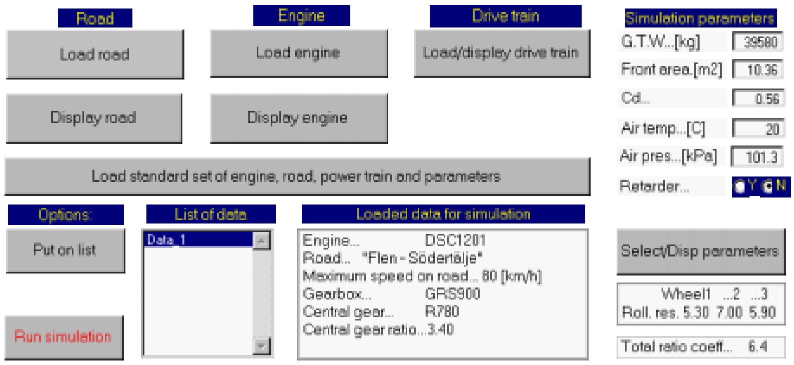

Figure 2.2: User interface of Scania Truck and Road Simulation (STARS) application.

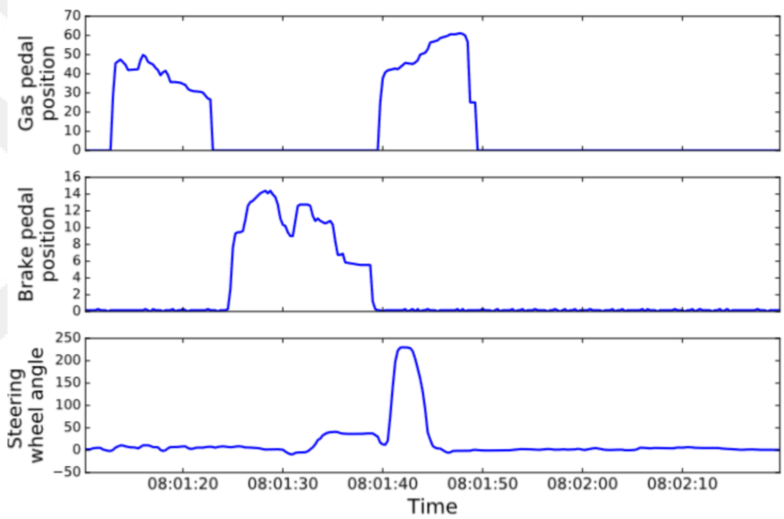

Figure 2.3: Example of signals acquired by the sensors synchronously.

To avoid confusion ,they developed a session based structure for each user using the cars. The sampling is not uniform due to the particular characteristics of the Can Bus and the signals. They sampled continuous signals at 20 Hz high frequency while low frequency sensors reports their data only when there is a change in their value (e.g. rain sensors,seat belt sensors, etc.) but for the sake of simplicity all the signals considered in their analysis, were re-sampled at 4 Hz through linear interpolation. 8 different features were chosen (Brake pedal pressure, Gas pedal position, Revolutions per minute, Speed, Steering wheel angle, Steering wheel momentum, Frontal acceleration, Lateral acceleration). After extraction some features such as values of the signal for each sample, values of the local maximum etc. clustering algorithms were used to obtain % 99 impairing clustering performance.

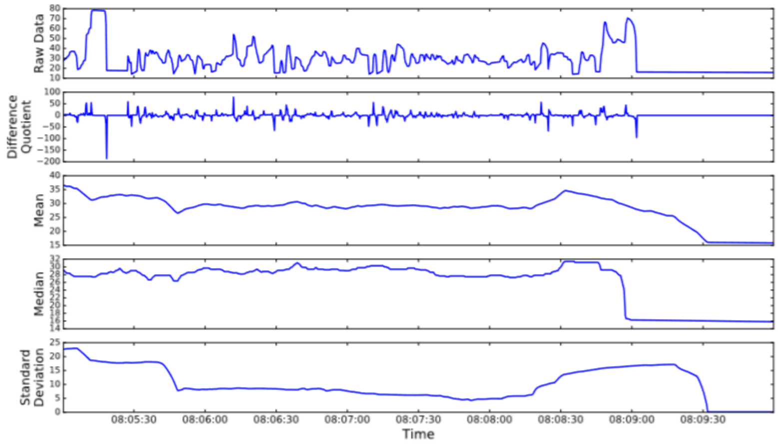

Figure 2.4: A sample of some of the features extracted from the gas pedal signals and its difference quotient, mean, median, and standard deviation.

The authors of (Jakobsen et al., 2013) listed a number of advice for eco-driving that will reduce fuel consumption. They analysed some well known advice such as maintaining steady speed, decelerating smoothly, avoiding idling etc. for all the advice, they evaluated them individually. Their data set contains records from four real-life diesel vehicles of minibus type. All vehicles were assumed to be comparable based on statements from the data provider. About % 99 of the data is recorded with 1- 2 Hz frequency while the remaining is collected at lower frequencies. They disgarded the last % 1 of low frequency data because high frequency data is needed in parts of their analysis. GPS and Can Bus data are used their work. Fuel consumption is often available in different formats, e.g. the fuel level in the tank, the instantaneous fuel consumption and the total fuel consumed. Instantaneous fuel consumption is estimated based on other Can Bus data such as RPM and fuel flow. According to authors this makes it a good estimate of fuel consumption at drive time, but it is too inaccurate at an aggregate level. The total fuel consumption is more accurate when looking at consumption over time. The acceleration can be available in the Can Bus data, but can also be calculated from the speed. The position of the throttle indicates how aggressively the vehicle is driven. We can summarize the method they use by experiencing and saving the changes they get. The main idea of this paper, Not one single of these factors has been found to lead to a good fuel economy on its own but several factors have been found where the least fuel efficient vehicle can improve i.e. "This vehicle idles much more than the others, and it drives at much higher speeds". On the contrary, it drives through fewer traffic lights. This leads to the conclusion that no single factor is all-important, however,

Vehicle id	Number of records	Start date	End date
1	1,847,170	2012-08-03	2013-04-30
2	1,917,610	2012-08-07	2013-04-30
3	938,068	2012-08-01	2013-04-30
4	1,862,091	2012-08-02	2013-04-30

Table 2.1: Data properties of (Jakobsen et al., 2013

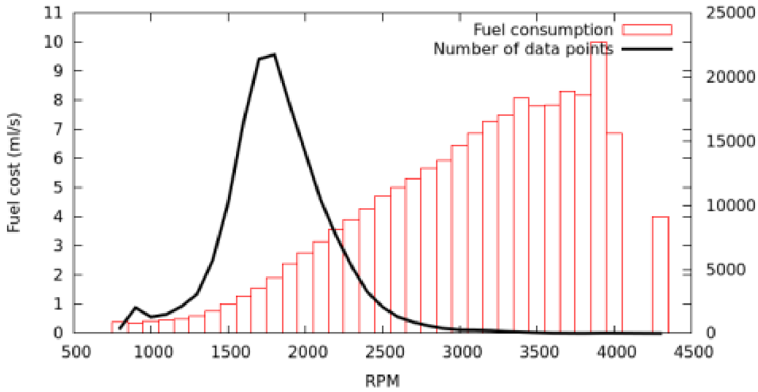

Figure 2.5: Average fuel cost per second when accelerating with varying RPM

it is the many small improvements that leads to a good fuel economy.

2.3 Machine learning based approaches

In this chapter, articles about machine learning and estimation of fuel consumption which is one of the targeted methods are examined.

In the study of (Zeng et al., 2015) they developed a mathematical model that estimates the trip fuel consumption by using machine learning approach. Various explanatory variables including trip distance, average travel speed, intersection density, coefficient of variation for link speed, and engine displacement, are captured by GPS and Can Bus data from 153 probe vehicles during one month. The ultimate use of their model would be incorporated into an eco-routing navigation system which provides reliable and environmentally friendly driving. They used Large-scale fuel consumption data and GPS trip records collected for a detailed analysis on trip fuel consumption rather than the instantaneous fuel consumption. Considering various factors, a support vector machine was applied to model the vehicle fuel consumption for a trip. They also applied sensitivity analysis and elasticity analysis for each of the explanatory variables are conducted to reveal the relationship between inputs and output of the SVM model. To predict for trip fuel consumption they used GPS data for calculate distances for using model target. Their experimental results show that they proposed SVM model considering the feature of the whole trip is more accurate than that of the Lineer Regression model based on the summation of link fuel consumption.

Although the Can protocol is standardized, the actual implementation differs for every manufacturer and even differs for every model. Automotive manufacturers are free to choose which ID represents which data, how the data is represented, etc. This configuration is not publicly disclosed. This means that Can Bus traffic for every car has to be analysed and reverse engineered in order to obtain useful information for a find general solution of instant fuel consumption. In the paper of (Huybrechts et al., 2018) , they propose an automation of the analysis steps of reverse engineering in order to improve and facilitate the process. Two approaches of automation are discussed, namely an arithmetic approach and machine learning using classification. In conclusion, they show that reverse engineering of Can Bus traffic is at least partially possible by applying machine learning techniques and the performance of the classifiers increases when adding additional features to the other analysis. In the arithmetic approach, they performed multiple test drives while recording the Can Bus and diagnostic information from the OBD connector. The requested data from the OBD contains the throttle position, vehicle speed and engine RPM. They collected ten best results from the RMS method and tried to understand with ID could be represents original one, so visual inspection is required to verify the results of their arithmetic approach. On the other hand for the automatized way to identify them, they used machine learning models such as SVM, Logistic Regression, LSTM etc. Their results show the possibilities to, at least partially, identifying signals of interest in the data of Can Bus traffic. Additionally, they have proven to increase the performance of certain classifiers used. However, more experiments on larger datasets and different vehicles need to be performed in order to validate their findings. There is not a general solution for identifying Can Id's automatically.

There is another application of (Perrotta et al., 2017) and they modelled fuel consumption prediction of a large fleet of trucks and test the use of telematic and road condition data from fleet managers and road agency databases. They applied SVM, RF, and ANN techniques in modelling part and compare the performance of the fuel consumption of large truck fleets using the data which come from sensors installed on the most recent trucks as standard. For their study, anonymized data was provided by Microlise Ltd, a company providing telematics and truck fleet management services. Only articulated trucks are considered in their study. It was possible to identify each vehicle only by an ID number in reference to its system of sensors, the wheel configuration, and the type of engine. Date, time and an unambiguous ID number identifies each of the records. This information allows the journey of each vehicle to be retraced. In order to define their target variable, Fuel consumption was calculated as the ratio between the fuel used and the distance travelled, transformed into l/100km, so they did not predict the instant fuel consumption. This may lead to their conclusion that RF was the best technique to predict fuel consumption, however, the SVM and ANN demonstrated a good level of accuracy and in particular they can be considered more accurate than the RF in predicting extreme values.

In Remeli et al. (2019), utilize Can Bus data in its raw format, byte by byte without resolving the ID's or separating the bits of the can messages in order to perform operator prediction. They train a deep learning model using can bus data and are capable of determining the driver of a vehicle (among 33 drivers) with %85 accuracy. Although this approach is relatively effortless, a similar approach and model would not fit our needs only estimating the fuel consumption of a vehicle would not enable the amelioration or optimization of the causes for lower consumption. In addition on the test that we performed on operator prediction using raw Can Bus data, our system produced extremely high accuracy's with randomly selected lines of data. It was later discovered that that the model was utilizing the time code and seat belt information in the can bus to determine the operator from the training data.

In the studies mentioned above, the fuel consumption information consumed by the total driving journey is modeled or analyzed. In the method we are trying to contribute with this study, we try to provide the factors that can reduce fuel consumption more effectively by modeling the instantaneous fuel consumption information that may affect hundreds of inputs.

	CAN	CAN+GPS	CAN+greedy	CAN+GPS+greedy
Discriminant Analysis	64.9	65.6	71.9	73.0
K nearest neighbours	63.1	68.6	76.4	76.4
Logistic regression	68.8	67.3	74.2	78.1
Naive bayes	66.5	66.4	71.9	71.9
Support Vector Machine	64.2	69.8	74.0	78.2
CNN	91.0	93.8		
LSTM	90.8	87.7		

Table 2.2: Overall Performance of the Different Classifiers for ML approach in reverse engineering on CAN bus data

Model	RMSE	MAE	R2
Support Vector Machine (SVM)	5.12	3.56	0.83
Random Forest (RF)	4.64	3.21	0.87
Artificial Neural Network (ANN)	4.88	3.46	0.85

Table 2.3: Model results for the Application of fuel consumption modelling on trucks

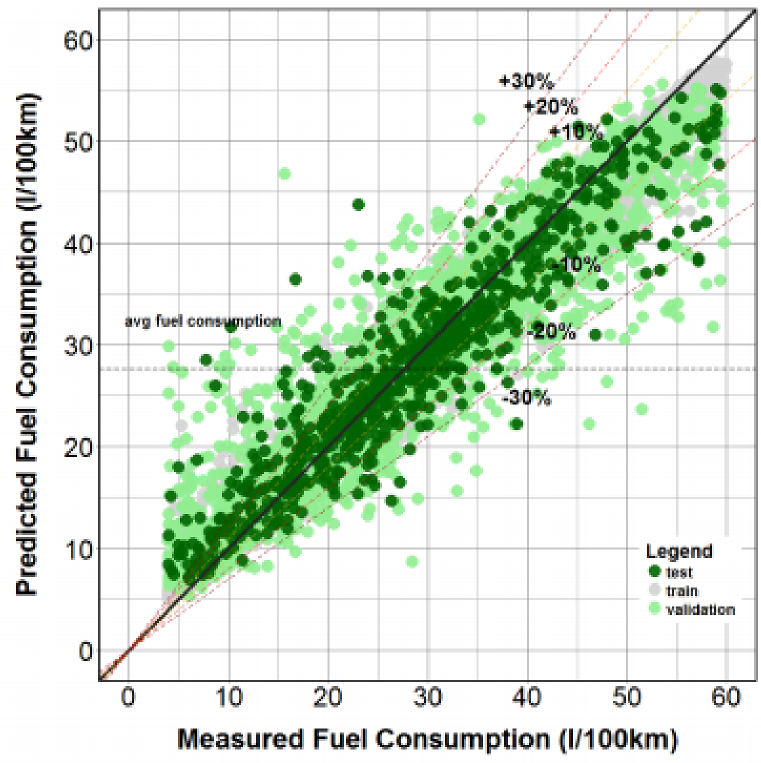

Figure 2.6: Fit of the developed RF

3 METHODOLOGY

The data collection setup for the project consists of a Raspberry Pi 4 with canable CanBus transceiver. Additional sensors such as a GPS receiver and an accelerometer were incorporated in the original designs, however the vehicle already has an accelerometer so they were not added to the project. When the vehicle is powered, the systems starts collecting data.

After the process for compiling the data set, training the models and evaluating the results can be broken down into a data mining phase and a training phase. The data mining phase is the collection, analysis and consolidation of data from the different sources. The training phase is using the consolidated data and fitting ML models to it. Once the training is finished the results will be evaluated to determine which features are most influential and which model performed best.

In the data mining phase, LSTM networks are be applied for built models. Long Short Term Memory networks usually just called “LSTMs” – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by (Hochreiter & Schmidhuber, 1997), and were refined and popularized by many people that can follow Felix Gers (2015). LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information for long periods of time is practically their default behavior, in other words not something they struggle to learn.

3.1 Connecting to the CAN BUS

This section describes our experimental setup to connect to the CAN BUS.

At the beginning of the data collection, the OBD2 protocol was evaluated. The

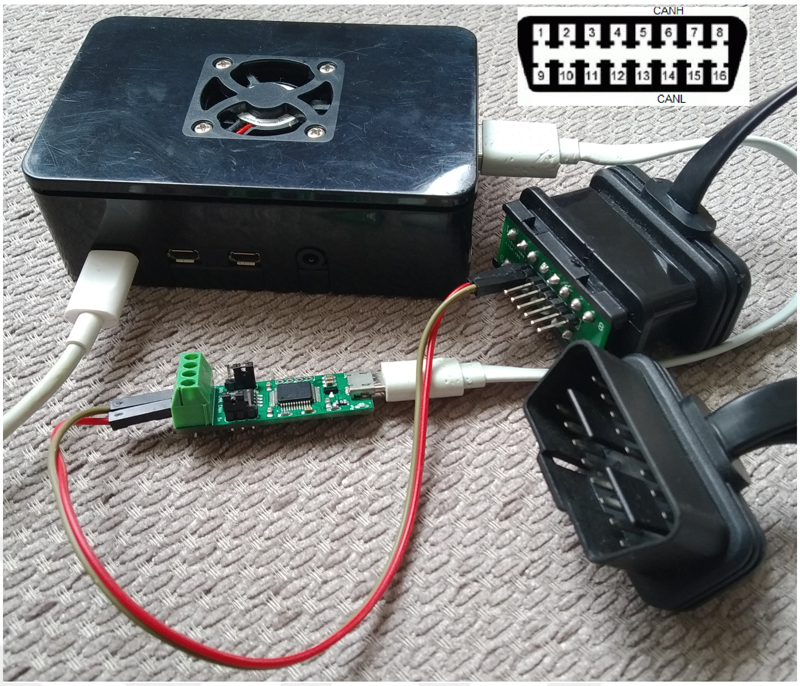

Figure 3.1: Can Bus Logging System

OBD2 (On Board Diagnostics 2) protocol is a vehicle/manufactuerer independent protocol that is mandated on all vehicles. It provides data over the Can Bus. Alghouh the OBD2 protocol does provide a lot of useful data that is well documented, the speed of the protocol was not sufficient to acquire data at the desired frequencies. Utilizing the OBD2 protocol would have yielded one or two data points for the 0 to 100 km/h acceleration of the vehicle. In addition the protocol did not provide the fuel consumption figure for the test vehicle Chevrolet Aveo 1.3 Diesel (2012). So we focused on reaching the Can Bus data directly. To achieve our goal, we used a specific device which is called CANable. The CANable is a small low-cost open source USB to Can adapter that shows up as a virtual serial port on your computer and acts as a serial-line to Can Bus interface. It also has python library support for data logging operations. To access the Can Bus we connected CanHigh and CanLow lines on the vehicles diagnostic (6 (Can H) and 14 (Can L)) socket to the respective connections on the Canable device.

We designed a data logging script with python for logging process and ran this code in a Raspberry 4 sigle board computer (SBC) which is directly connected to the car's Can Bus line with CANable device. The device is powered from the vehicle's 12v input. Thus, the vehicle continuously collects data from the moment it starts working until it stops. More than one hundred trips were collected by three different

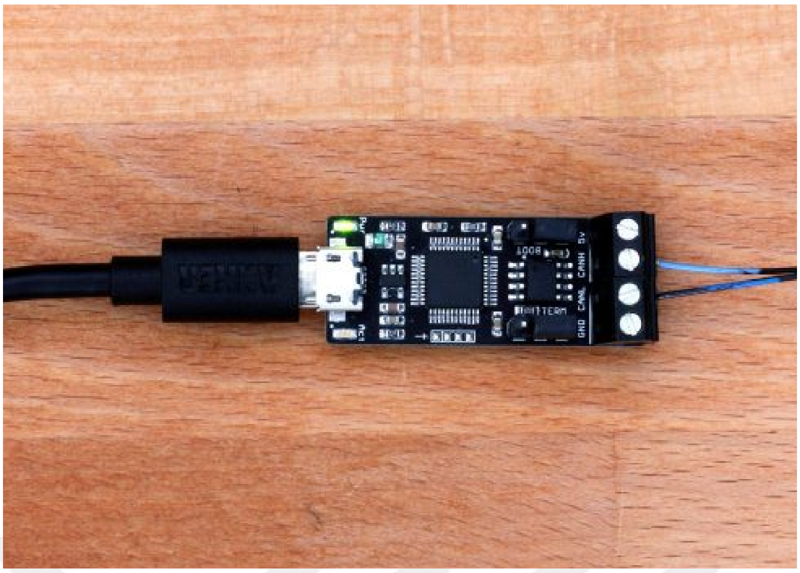

Figure 3.2: Canable Device

drivers over a 9 month period starting in October.

Thermal management of the SBC was accomplished by means of aluminium heat sinks over the CPU and forced air cooling using a fan on the SBC case.

3.1.1 Formatting Sequential Data

Python-can is a python library that allows easy communication on the Can Bus using Python. The library supports connecting to Canable/Cantact devices directly with via a serial connection on Windows or Linux and also can directly work with socketcan devices on Linux with the candlelight firmware.

The reason for the sleep method, which is in the early stages of the code, is that when the vehicle's engine starts, a certain time must pass before the code can work properly. We have added the relevant block to the code to solve this problem. All the data read from the Can Bus line is recorded as an array containing the Can ID, date code, and data received from the bus. This is later logged on to a SD card in csv format.

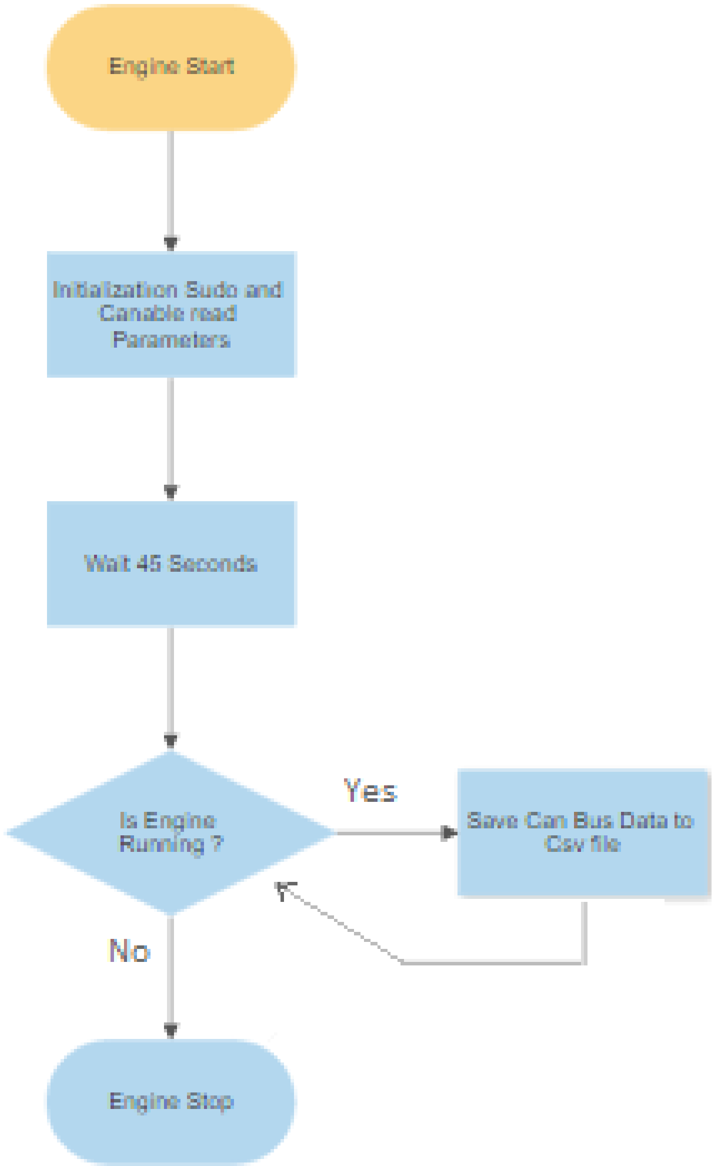

Figure 3.3: Can Bus Logging Flow

time	id	dlc	data
21259.000000	0x1C6	8.000000	[0, 0, 0, 0, 0, 67, 255, 255]
22651.000000	0x1E1	7.000000	[0, 0, 0, 0, 1, 20, 224, 0]
23980.000000	0x1F3	3.000000	[0, 60, 0, 0, 0, 0, 0, 0]
26415.000000	0x1E5	8.000000	[78, 247, 203, 224, 4, 8, 50, 0]
27555.000000	0xC1	8.000000	[176, 108, 44, 96, 112, 107, 49, 59]

Figure 3.4: Data Format

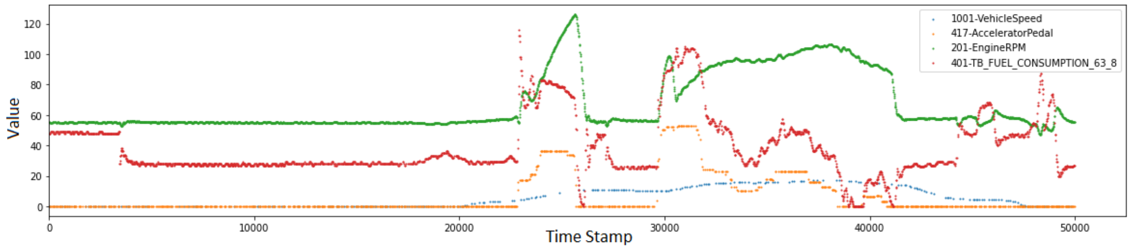

Figure 3.5: Detect Fuel Consumption Information

3.2 Reverse Engineering

Without manufacturer collaboration, decoding the information provided by the vehicle is a very tedious task and a considerable amount of time was devoted to determining the meaning of the signals of the Can Bus. The messaging protocol was partially reverse engineered using the following methods.

3.2.1 Obtaining the Fuel Consumption Data

In this section we document our methodology to obtain the fuel consumption information.

To find out the target variable which is instant fuel consumption information, we developed a machine learning model which is trying to predict the gas pedal information. As an expert opinion, fuel consumption and gas pedal information are strongly correlated to each other. After building the machine learning model, we collected feature importance values of the all variables and removed all the non related features one by one based on the variable plots.

3.2.1.1 Finding an Entry Point

As there are 66 different message ID's on the Can Bus and each message contains up to 64 bits of data in no particular order. It was not obvious which ones would be of interest. The task of decoding each and every one would be a task that is far too time consuming and difficult for the scope of this thesis.

As applied in the article (Pheanis & Tenney, 2003) , the message ID's were classified according to their standard deviations and also their frequencies with the assumption that higher frequency ID have a larger importance in engine management. In addition low standard deviation data (such as seat belt engagement information) was tagged to be examined in later steps. Data which is always 0 was also marked and not taken in to consideration in our dataset.

The approach adopted in this work is to find the fuel consumption information and and utilize all other data that is unknown be utilized by the machine learning algorithms.

Finding and entry point in to this maze of unknown data proved not to be an easy task. Probably the most obvious influencer of fuel consumption is the accelerator pedal of the vehicle. The advantage of trying to utilize the accelerator pedal for entry is that its values change even when the engine is not running.

Visually inspecting the data revealed several bytes of several ID's that moved with the gas pedal in addition to values that moved inversely with the accelerator pedal. To determine which ID's and values on the Can Bus refer to the accelerator pedal and prediction algorithm. We used XGBRegressor for predict gas pedal position. In this way, the variables of the highest order of importance for the relevant ID were examined.

This method revealed 5 ID-byte combinations, plotting the data all of the combinations had the same value.In order to find the most appropriate value to be utilized for the following sections of this study, it was important to determine which one of these is the primary source of information and is first put on the Can Bus by the module reading the accelerator pedal, and to which ID it is transmitted first, rather than other modules that could be echoing the information they receive. The Can Bus signal that first appears on the Can Bus also had the lowest ID, indicating that it has a higher priority.

In order to find the most appropriate value to be utilized for the following sections of this study, it was important to determine which one of these is the primary source of information and is first put on the Can Bus by the module reading the accelerator pedal, and to which ID it is transmitted first, rather than other modules that could

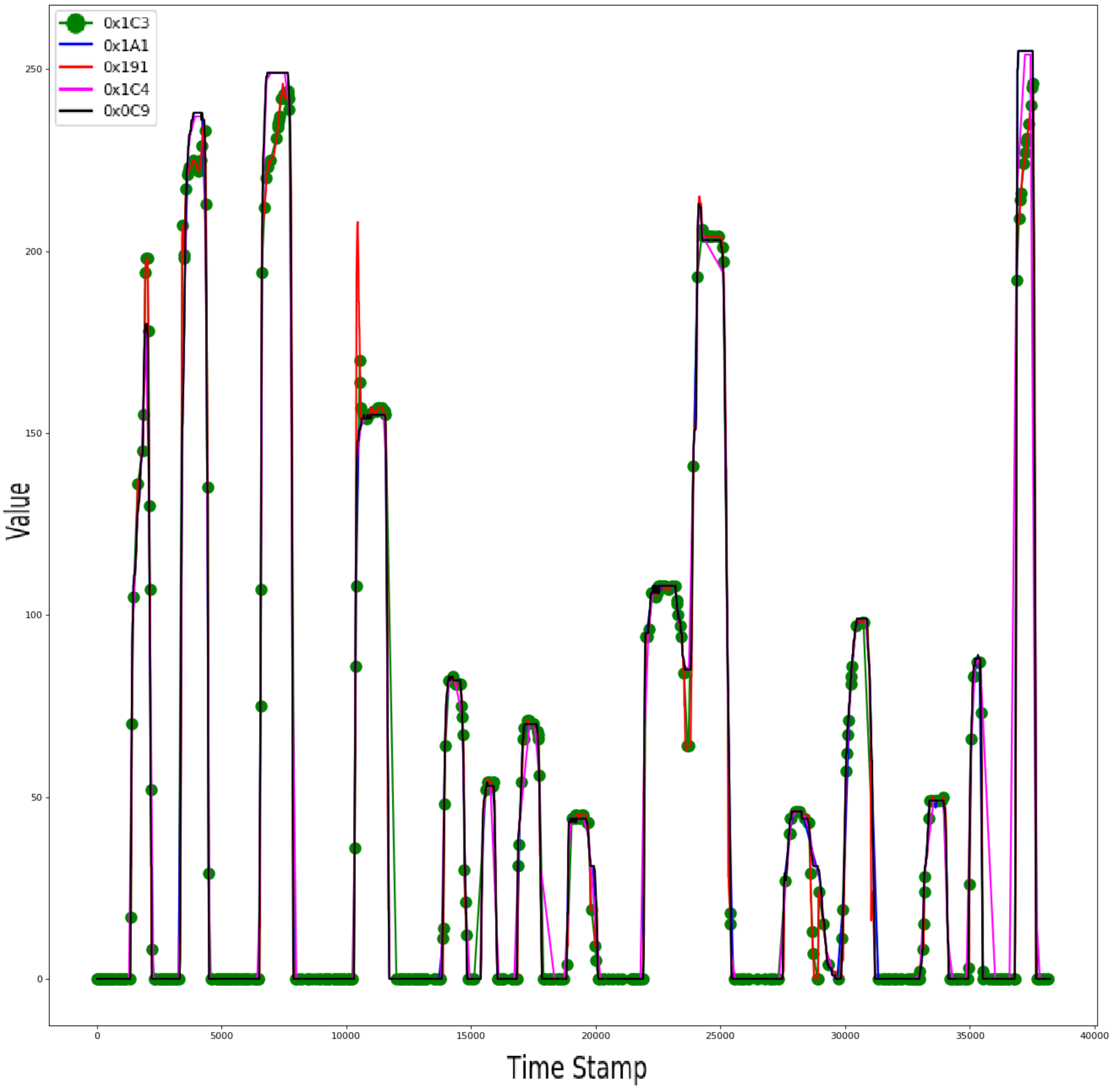

Figure 3.6: Gas pedal CanIDs for total journey

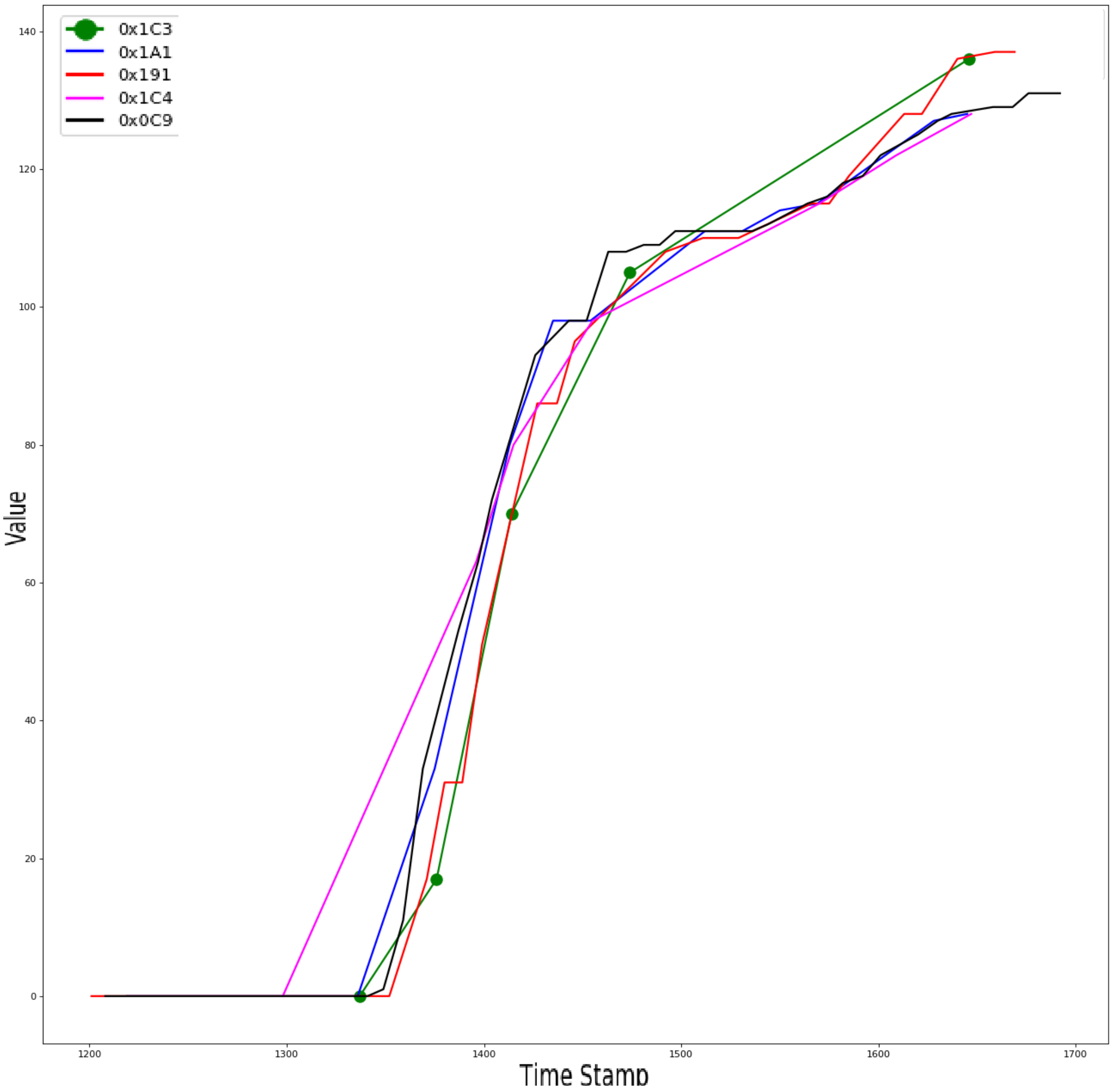

Figure 3.7: To Understand signal order

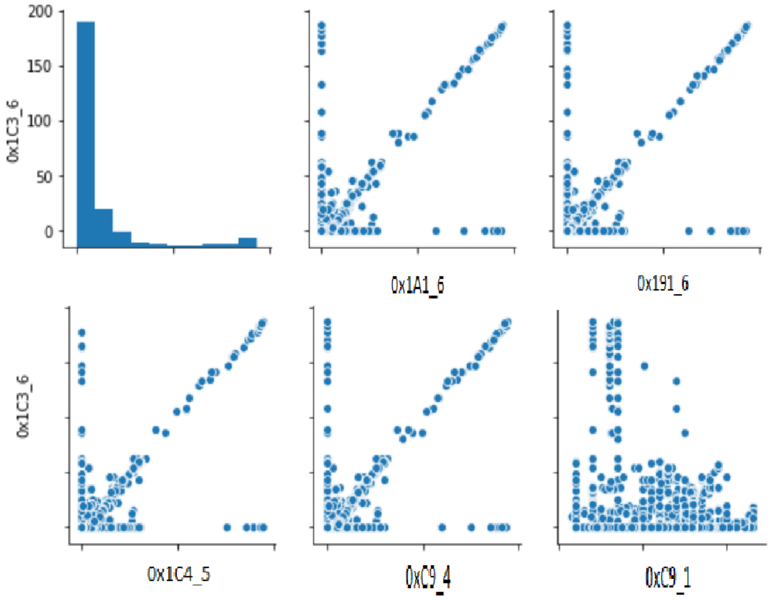

Figure 3.8: PairPlot Graph for Analyse Reletad Variables

be echoing the information the receive.

Plotting all of the values obtained it was concluded that ID 0X9C byte 6 is the first time that the gas pedal information appears on the Can Bus.

3.2.1.2 Obtaining Fuel Consumption Related Information

In this part of the work, our assumption is that the gas pedal position has an effect on fuel consumption. Therefore if we train a model to predict the accelerator pedal position, one of the fields with the strongest influence in the vast amount of data provided by the Can Bus should be our fuel consumption.

After obtaining the accelerator pedal position with the engine off data, all related ID's that mirror this information and its inverse are ignored as this information would be utilized to predict the accelerator pedal position.

After this phase, data is collected with the engine running and the vehicle in a stationary position. A new model to predict the gas pedal position is created and trained on this new dataset.

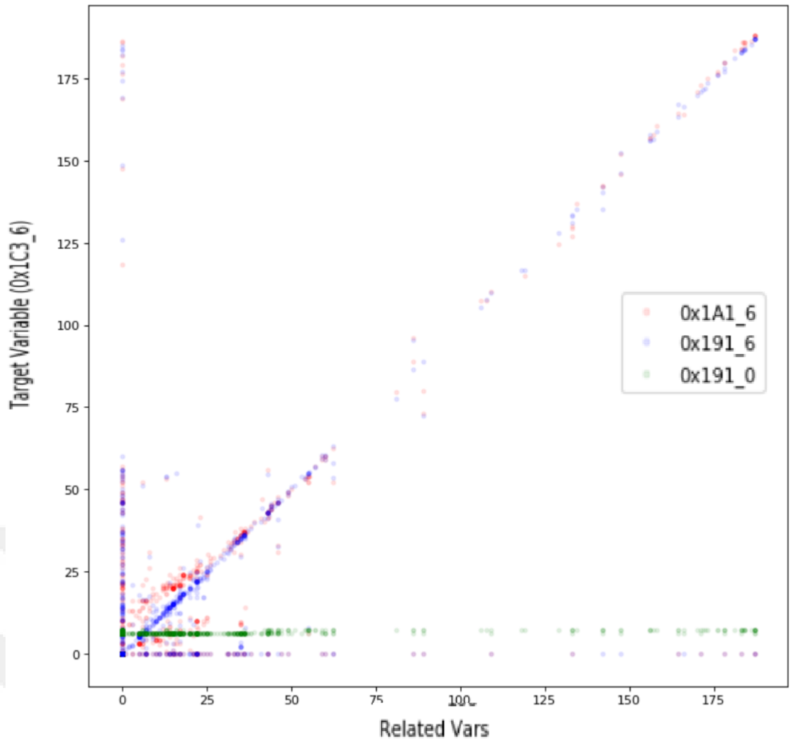

Figure 3.9: Scatter Plot Graph for Analysis Related Variables

3.2.1.3 Obtaining Mass Air Pressure, Mass Air Flow and Intake Air Temperature Information

It was important to determine which ID's contain information such as intake air temperature, mass air flow and mass air pressure sensor values. These values have a great importance and direct effect on the amount of fuel injected. If we want to establish an understandable and accurate prediction model, we must identify these values.

To achieve this aim, we collected five different datasets which have different characteristics. In the first stage, we started by exploring the location of the sensors in the vehicle and investigating how we can access them. After consulting the service manuals disassembly instructions, we collected data by removing the MAF (Mass Air Flow) sensor of the vehicle and blowing air to the sensor sequentially with a hair dryer and leaf blower. This process was repeated with the vehicle engine not running and running. In this manner, we tried to visually correlate the sensor output and the corresponding Can Id..

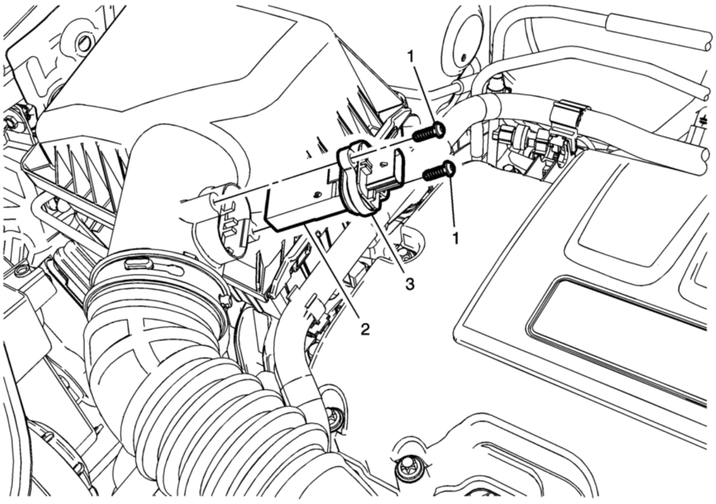

Figure 3.10: Maf Sensor Location on Chevy Aveo

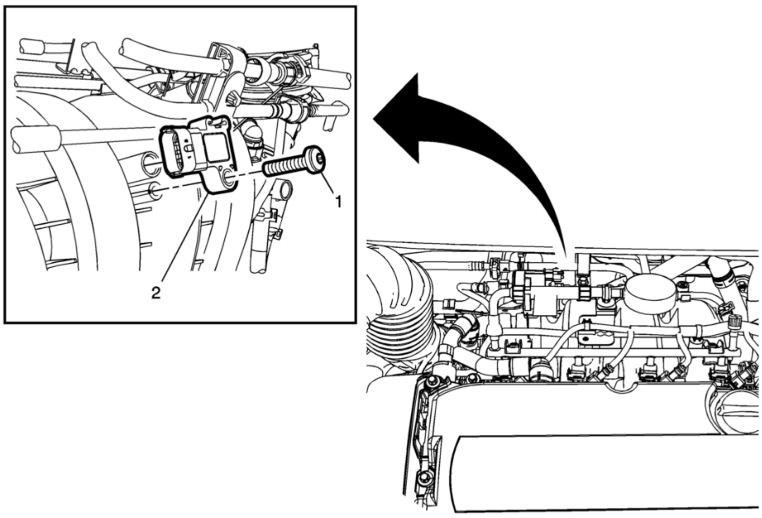

Figure 3.11: Map Sensor Location on Chevy Aveo

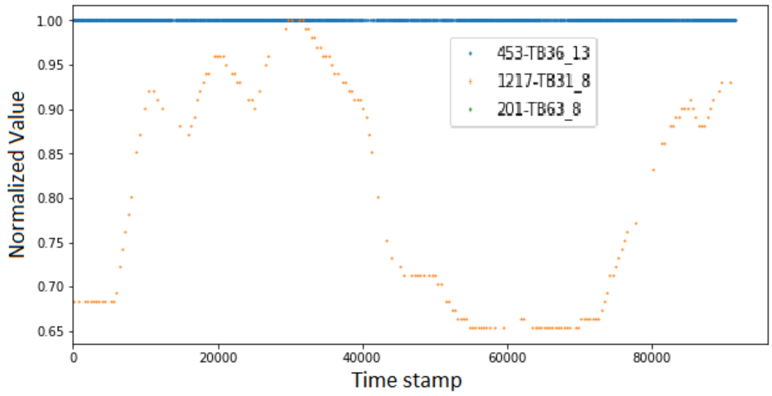

Figure 3.12: Hair Dryer Effect On Engine Close Statement

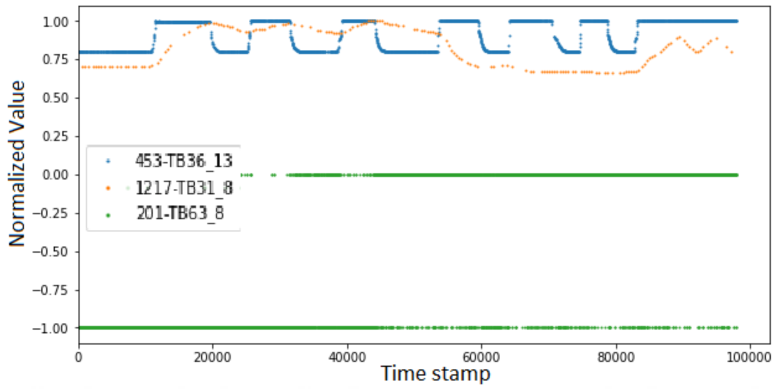

Figure 3.13: Hair Dryer Effect On Engine Run Statement

To identify MAP signal, we collected similar data from vehicle but this time the vehicles intake pipe was disassembled after the air filter and the leaf blower was attached with an airtight seal. This experiment was repeated with the engine running and not running which similar to the previous experiments. After restoring the vehicle, we finally gathered one more data while the engine was not running. We aimed to use this data to interpret the results we found more accurately.

After completing the data collection phase, we plotted all the signals in our database. Then we collected all signals that have changes. To understand Intake air temperature, We repeated the same process at short intervals by blowing hot air to the MAF sensor 3 times in certain periods and then blowing cold air. When we examined the graphics, we came to the conclusion that the signal named 1217TB3613 contains IAT information (orange line in the graphs). When we examined the 453TB3613 signal (blue line in the graphs), and determined it to be the MAF information in our data. The number of dislocations in the chart and

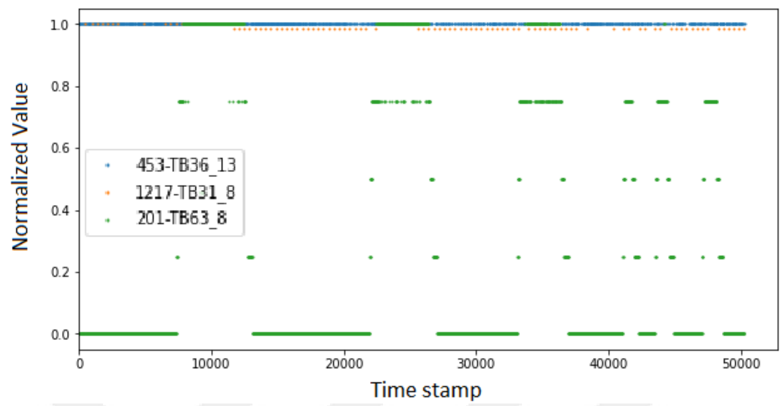

Figure 3.14: Leaf Blower Effect On Engine, Engine Not Running

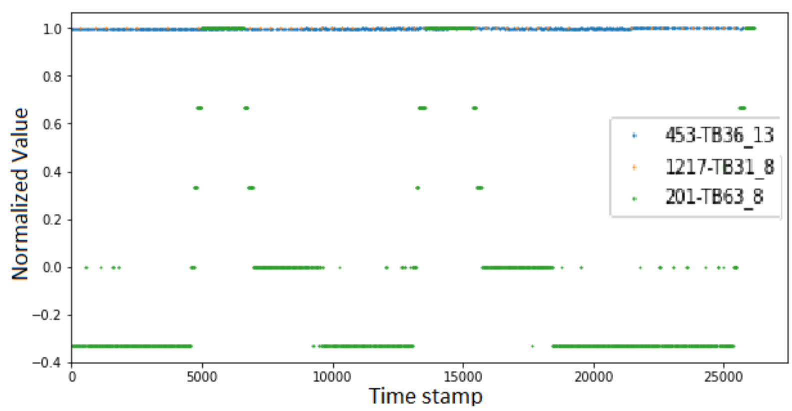

Figure 3.15: Leaf Blower Effect On Engine, Engine Running

our experiments in the data collection phase match. Another comment we can make on this subject is that this information is produced only when the engine is running.

The signal representing the MAP information is 201TB638 which is a green line. When we look into engine running statement. We can see the negative pressure on the graph. Such a situation is not observed when the engine is not running and this makes sense to us. Also we can detect the number of blowing events with the leaf blowing machine on the graph.

3.2.1.4 Determining the Fuel Consumption Field

In this section, we describe our effort to determine which of the reported values could be the fuel the consumption information.

3.3 Predicting Fuel Consumption

After the fuel consumption value reported by the vehicle is obtained (or derived using injection time and rpm), a model is formed to predict fuel consumption information.

As speed, air temperature and lots of other factors have an effect on fuel consumption, a vast amount of data is collected to train this model. In order to apply the machine learning approach, we need to pass the inputs we have through data pre-processing steps. First of all, Line-based data consisting of message ID's are transformed into signal-based variables by *pivot* operation based on index value as an group by option. At this stage, since the frequency of sending some ID's in Can Bus is different from each other, there will be no data corresponding to each index value grouped, so the data starts have missing fields. We completed this missing missing information with the last data we observed on the bus. Pandas $fillna$ function with $ffill$ parameter was utilized to achieve this. In order to make this more effective, at the first stage, we start by creating a dummy dataframe of zero values for each message ID's and combining them with the data to which applied a pivot operation.

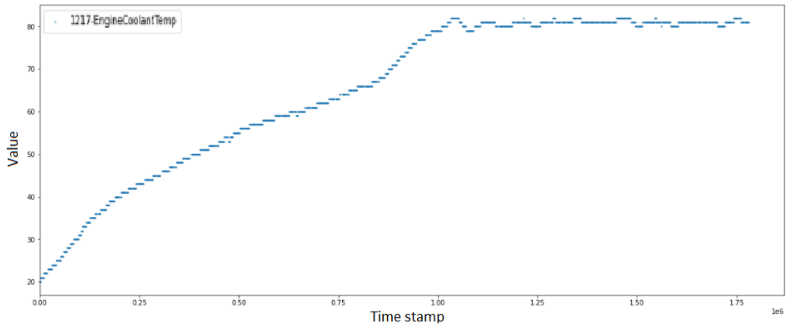

Figure 3.16: Engine Cooling Temperature Values of Train Data

After these processes, we tried to estimate our target variable that we have determined in the reverse engineering stage with all the all inputs we have. At this stage, we realized that adding all the inputs to the model causes some problems. Firstly, there may be variables that can be directly correlated between inputs and our target variable. So just like the gas pedal information there may be more than one message ID's that indicates the fuel consumption what we can not detect in the reverse engineering step. If we had not realized this situation, our model would utilize redundant information to perform its prediction. To overcome this problem, we experimented with input combinations iteratively, examined the variables that directly explain the model with a high importance, and removed the illogical ones from the input list. In our experiments, we have seen that inputs such as accelerator pedal related message ID's and Throttle position sensor information are directly correlated with fuel consumption and we have removed such variables from our input list.

The second problem of modeling with all inputs is that in the next phases of this project we shall implement real-time forecasting on the Raspberry Pi, so the estimates must arrive in a timely manner. Therefore, over complicating the model with unnecessary inputs would hinder operations on the SBC which has limited processing power compared to its desktop counterparts. As a solution to this problem, we proceeded by finalizing our model utilizing only the first 20 most significant variables describing our prediction model. In this way, we gained 80 percent of the scoring time and 90 percent in training time. (5 minutes to 30 seconds) Thanks to this time improvement in training time, we were able to spend more time on parameter optimizations and also produced more interpretable models by selecting less inputs. After all these improvements, 0.939 R square and 11,11 RMSE values were reached.

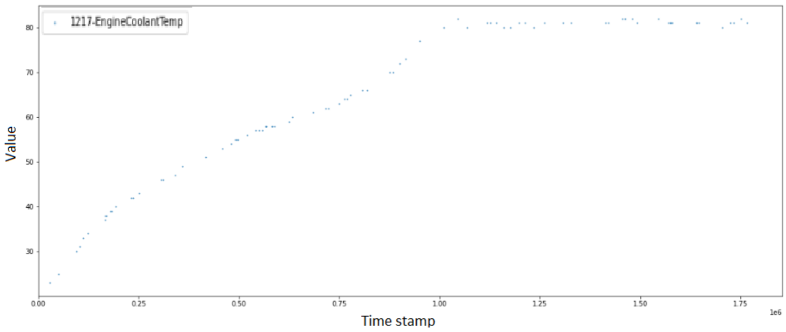

Figure 3.17: Sampled Engine Cooling Temperature Values of Train Data

3.4 Model Understanding

Due to the black box algorithm (Xgboost) which is an open-source software library that provides a gradient boosting framework. It initially started as a research project by (Chen & Guestrin, 2016) and it has gained much popularity and attention recently as the algorithm of choice for many winning teams of machine learning competitions that can follow (Xgb, 2020). We used that machine learning algorithm for generate accurate predictions. Applied many inputs that we do not know their possible effect to our target variable, as the mentioned (Lundberg & Lee, 2017) paper, the intelligibility of the model is of great importance for us. To achieve this, we used some model understanding libraries developed in python such as Eli5, Lime and Shap.

Eli5 is a Python library which allows to visualize and debug various Machine Learning models using unified API. It has built-in support for several ML frameworks and provides a way to explain black-box models (Eli, 2020).

Lime is a python library which tries to solve for model interpretability by producing locally faithful explanations. In the (Lim, 2020) page, that can find the Python API reference for the lime package

Shap (SHapley Additive exPlanations) is a unified approach to explain the output of any machine learning model. Shap connects game theory with local explanations, uniting several previous methods and representing the only possible consistent and locally accurate additive feature attribution method based on expectations. Beause of the local and global explanatory capabilities, Shap was the package we preferred

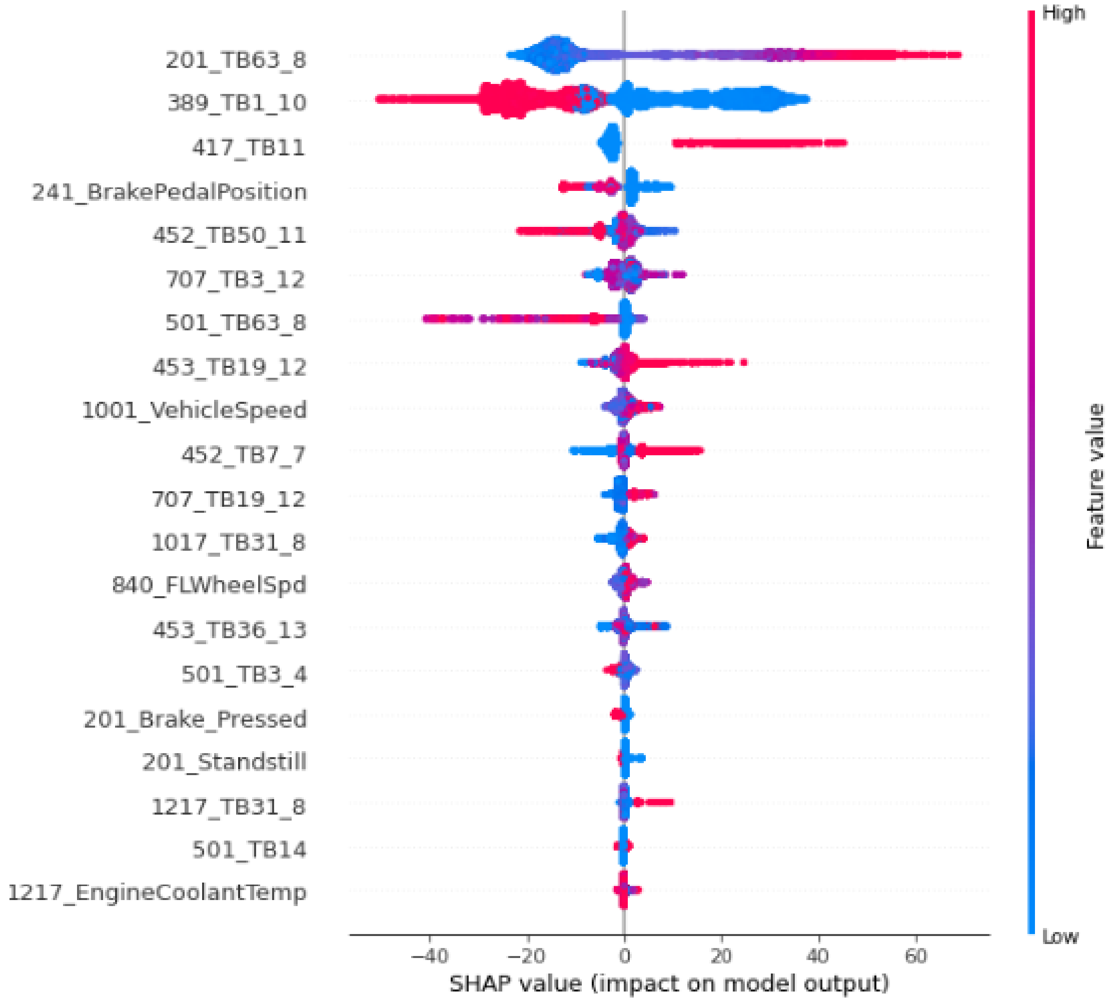

Figure 3.18: Feature Importance's of Prediction Model

to progress this work. In this manner, we were able to deduce and interpret how the inputs entered into the model and affected the predicted values.

When the model generating estimates, it is seen in which intervals the inputs produce high and low estimates (Lundberg et al., 2020). Red dots represent high estimates and blue dots represent low estimates 3.28. During the analysis of the variables that used in the model, we found that the order of feature importance of Xgboost and the order of importance of Shap were different. We found that the ranking formed by Shap is more logical. As an explanation for this effect, when we give the time value as an input to the model, Xgboost marks this as the most important variable, Because of the information gain function and tries to separate this information in its top nodes.

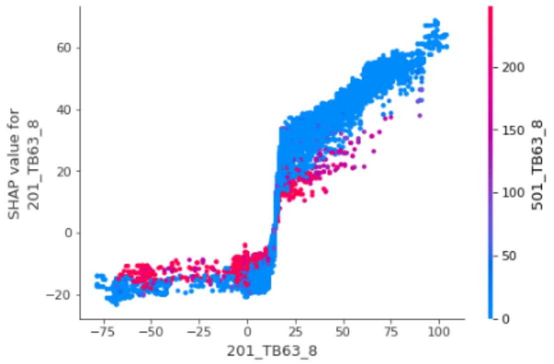

Figure 3.19: 201TB638

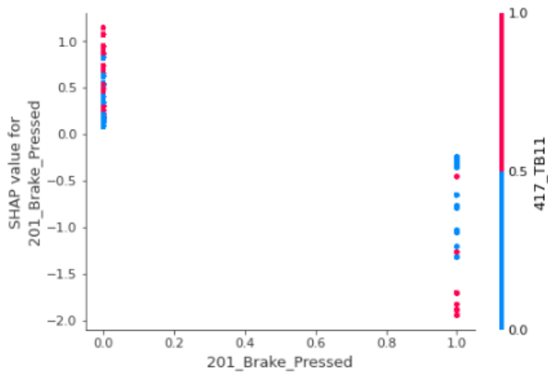

Figure 3.20: Brake Pressed

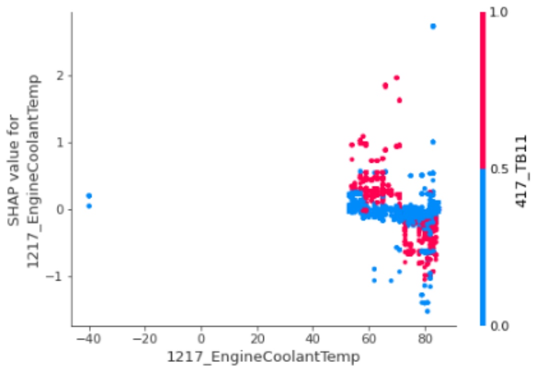

Figure 3.21: Engine Cooling Temp

Figure 3.22: Vehicle Speed

Figure 3.23: Model Inputs Understanding

It is also possible to visualize what kind of estimates the variables we use as inputs in the model produce for all estimates produced. As the mentioned in article (Lundberg et al., 2018) , we can summarize this as global explanation. you can see the global explanation of the model for different inputs.

If we examine local explanatory examples in , we can see how much inputs have an effect on the estimate generated. In this way, we can deduce how much estimation of the relevant model will produce in which input and compared to other variables for a single estimation. This topic is clearly explained in (Ribeiro et al., 2016)

3.5 Investigating the Effect of Engine Cooling Temperature on Fuel Consumption

In order to examine the effect of engine cooling temperature on fuel consumption, we aimed to create an ideal prediction model and change the model inputs to compare the predicted values of the system.

The main problem we experienced here was that engine cooling temperature information is not a significant input to the fuel estimation model. The reason is that information has not changed much in some of the data we have collected, therefore this type of collected data, does not effect our model outcome.

In order to find a solution to this problem, we examined all the data we collected based on the engine cooling temperature information. We chose the data which starts at low temperatures and reaches to 80-90 Degrees celsius (Engine operating temperature). You can follow in figure 3.16 for selected train data and our main information distribution. As the data we have specified contains millions of rows, we proceeded with sampling. The distribution of the data we sample is in the figure 3.17.

We created a fuel estimation model by following the modeling steps mentioned earlier. Estimation model with 5.2 RMSE value was obtained. You can see the estimation results and variable importance order of this model that we use the Xgboost algorithm. In this way, we have succeeded in establishing a model that can affect the fuel consumption of the engine cooling temperature information.

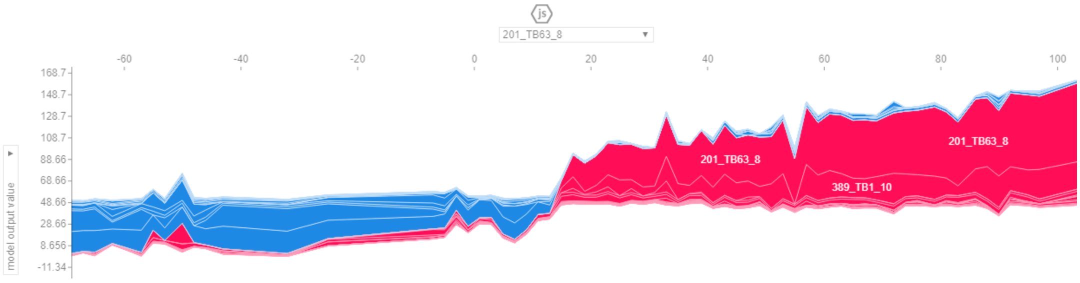

Figure 3.24: 201TB638

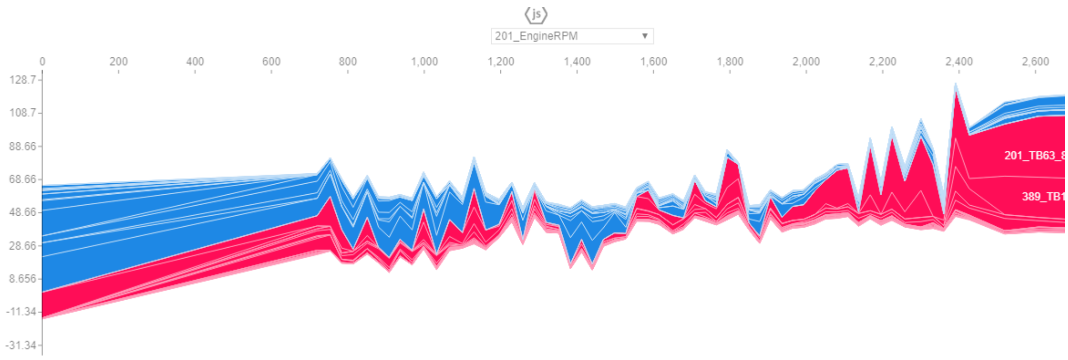

Figure 3.25: Engine Rpm

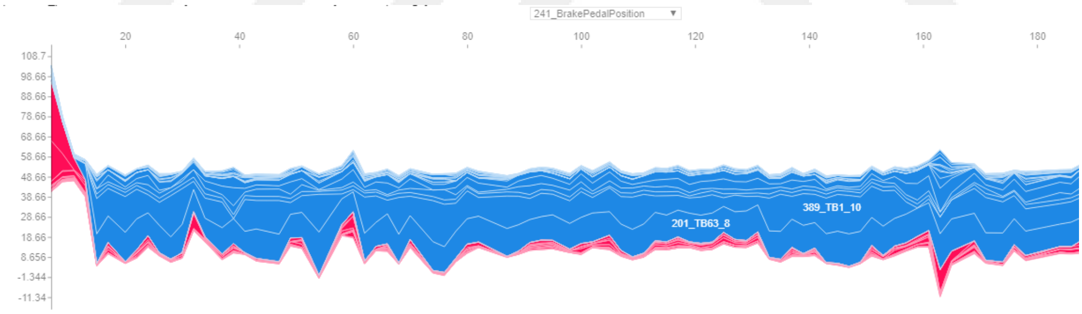

Figure 3.26: Brake pedal position

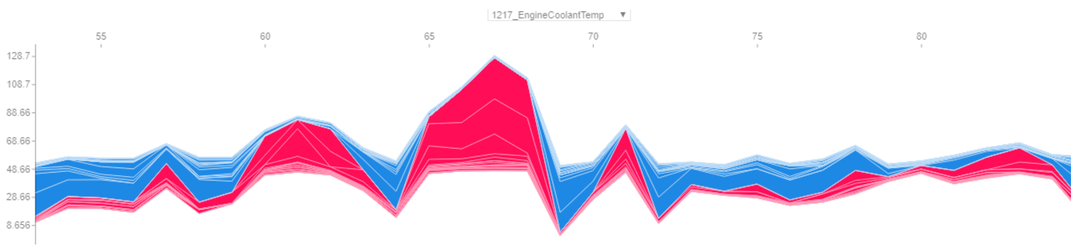

Figure 3.27: Engine Cooling Temp

Figure 3.28: Model predictions for all data based on input values for global understanding

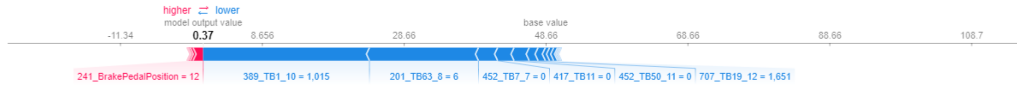

Figure 3.29: Low Prediction Examination

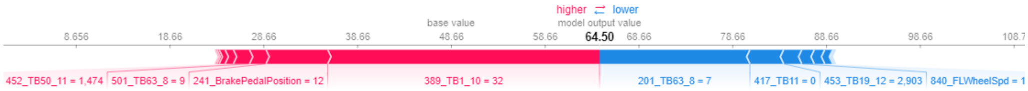

Figure 3.30: High Prediction Examination

Figure 3.31: Single Prediction Analysis (Local explanation)

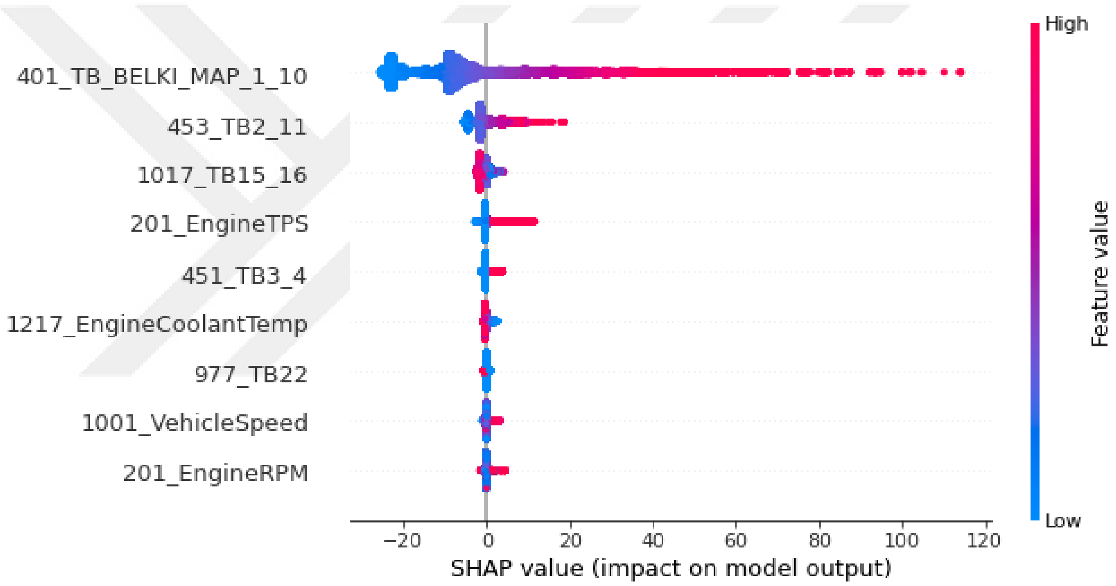

Figure 3.32: Model Importances for ECT Based Prediction Model

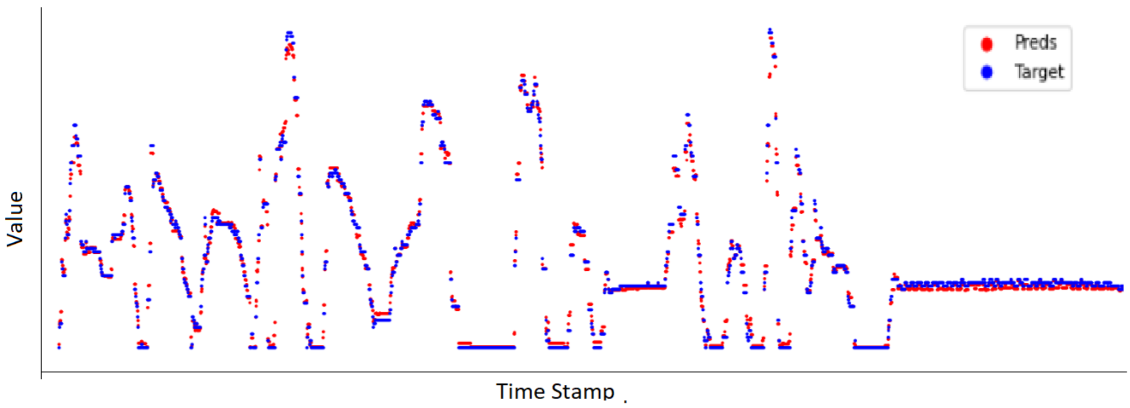

Figure 3.33: Prediction vs Target of ECT Based Prediction Model

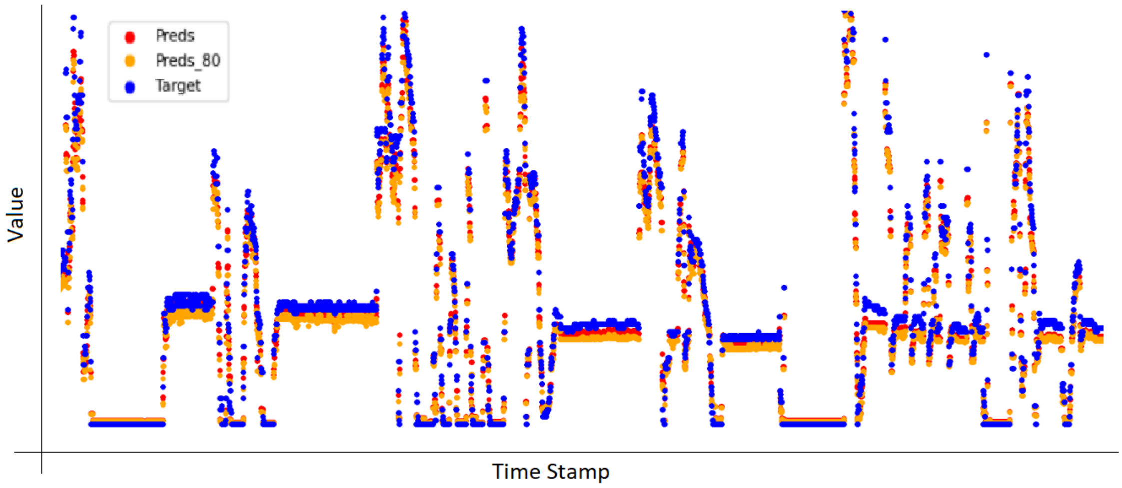

Figure 3.34: Engine Cooling Temperature Effect for fuel consumption

After installing our model, we predicted a different data that the model did not see during the training phase. In order to measure the effect of the variable on the outcome, we replaced the engine coolant information, which received different values, with the value of the engine's heated value of 80-90 centigrade, and predicted the model results for the whole data and as a result, we saw that, at values which the engine heated, The fuel estimation is less complete than when the engine is up to temperature. All of our comments can be viewed in figure 3.34. The orange values on the graph refer to the fuel consumption values that the engine could have received if it had been heated. The outputs obtained are included in the results section.

3.6 Investigating the Effect of Intake Air Temperature on Fuel Consumption

In order to investigate the effect of Intake air temperature on fuel consumption, we proceeded by taking samples from the data of many different journeys which has different outside temperature ranges.

We created the train data by taking a sample from the data ranging from 10 to 30 degrees Celsius. In this way, we thought that this variable could contribute positively or negatively to the model. Then, we made model experiments in an iterative way and finalized our variables.Although we did not contribute much to the model, we fed the information such as engine temperature and vehicle speed as input to the model as we will use it in our later analyzes. In this way, it became possible to examine the effect of IAT on fuel consumption in different input ranges.

The actual data on the Can Bus is coded with and offset of 40 degrees and coefficient of 1. In figure 3.37, orange points represent our IAT values without -40 offset value. So, the value of 50 appearing on the graph actually corresponds to 10 degrees Celsius.

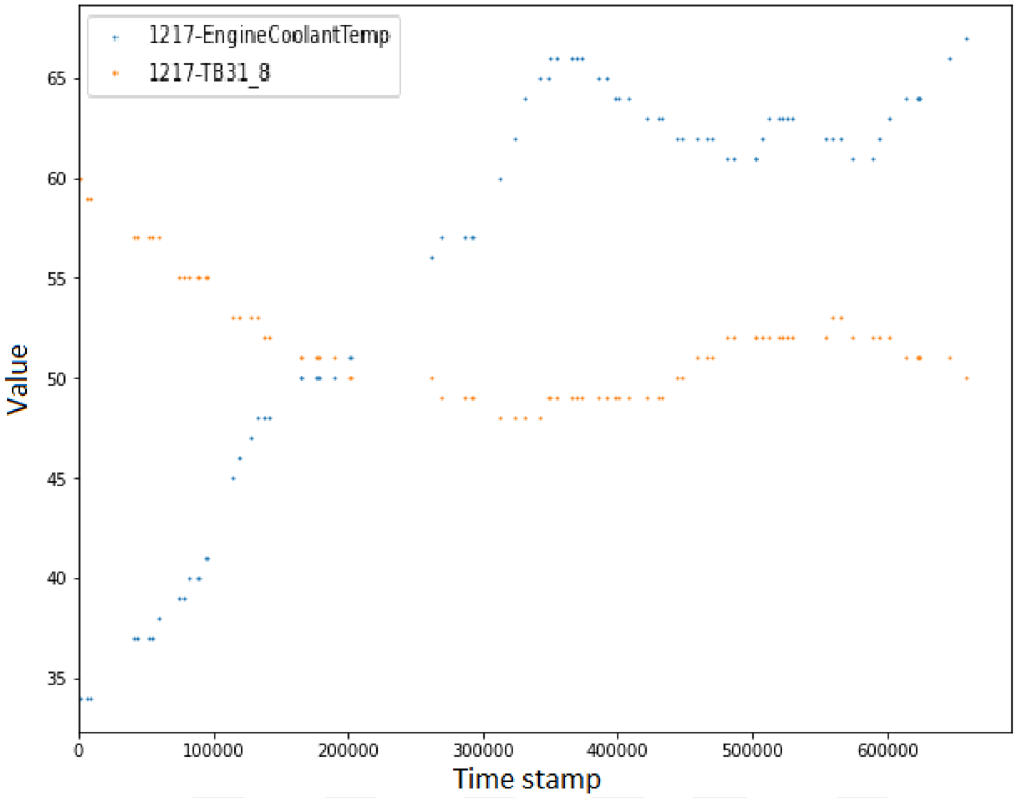

Figure 3.35: IAT Journey1 Example

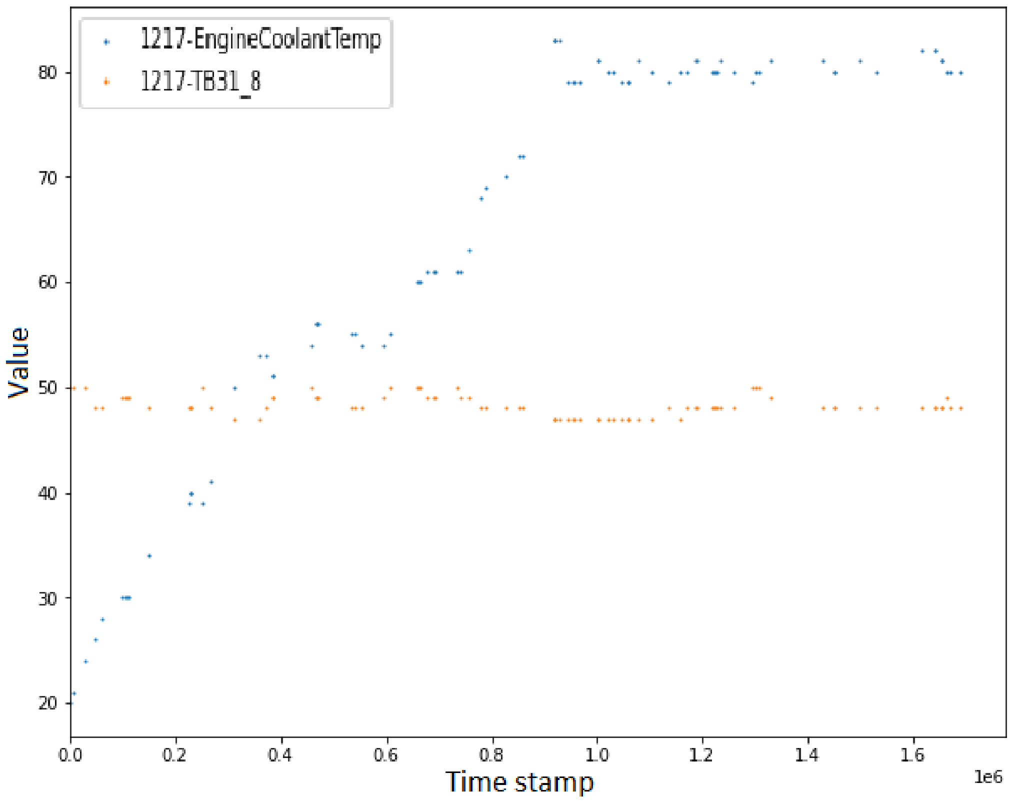

Figure 3.36: IAT Journey2 Example

Figure 3.37: Different Journey data for training phase)

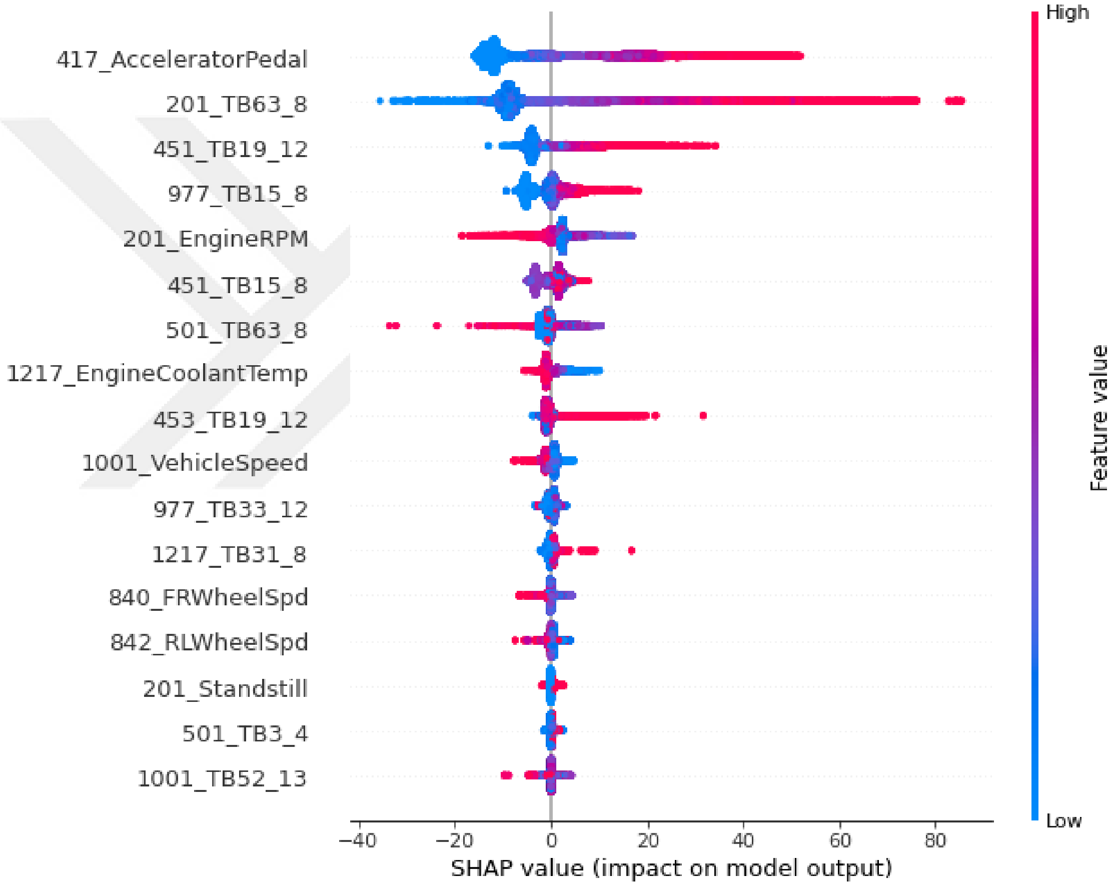

Figure 3.38: Feature Importances for Intake Air Temperature Effect Model

4 REAL TIME SCORING

The ability to score the models we have created with machine learning in real time as soon as the vehicle is running is a must for real-time actions such as on the fly changes to optimize fuel consumption. In the previous sections, we explained how we overcome this problem with limited computing and memory power. We use the 4 GB memory version of Raspberry pi 4 and this configuration is sufficient to generate scores in milliseconds time interval. Which is close to the data rate observed on the Can Bus.

When utilizing stored models some problems can occur when the software versions change. This can have a major influence on the output of the system. Correctly working systems can start to under or over report their outcomes. We build our model which utilizes version 1.0.2 of the Xgboost algorithm. To avoid the aforementioned problem, we utilized a library called joblib (job, 2020), which enables us to save our models as objects saved binary files. This enables us to transfer our models from the development environment to the deployment environment. This is especially important when the development and deployment are different, furthermore the problem can be amplified when there are different development environments such as servers dedicated to performing machine learning tasks in the cloud. An alternative method for avoiding this problem could have been strict synchronization of the versions among the different platforms, however the servers in the cloud platform were not fully under our control and were prone to perform automatic updates which created unforeseeable problems.

After installing the same version on the Raspberry, we transferred the scoring object to Raspberry thanks to the python joblib library. Model objects hold information such as feature names and model parameters. We use that information and design an adaptive scoring process. The models we created are variable name agnostic. This relieves us from the need to update the scoring code because the variable names change. This is a common occurrence in this project as more variables are identified they are given a name indicating their real world purpose. Therefore if we generate new model, we do not need to update our scoring code. This structure also enables us to update our model in real time i.e. improve the model on the fly while the vehicle is in motion. When we performed scoring tests, there was no difference in the scores. For this reason, there is no need for other methods such as Predictive Model Markup Language (PMML) which provides a way for analytic applications to describe and exchange predictive models produced by machine learning algorithms in Xml file format (Grossman et al., 1999). Our tests with PMML indicate that its performance is an order of magnitude worse than our models created using joblib.

5 EXPERIMENTS AND RESULTS

Having created a comprehensive model capable of predicting the fuel consumption of the vehicle we examine some situations and their effect on fuel consumption.

Our general model which takes in to account all of the input variables has an average error less than 1.2% error. However these results appear almost too good to be true. We concluded that our model might be mimicking the factory installed firmware of the ECU. This information is valuable however it may not provide an efficient means of evaluating different modifications to the vehicle. For example if there is increased drag or another form of inefficiency acting upon the vehicle, then the driver applies more throttle to bring the car up to speed or to provide the necessary acceleration to obtain the desired performance from the vehicle. Therefore we decided to omit from our dataset the driver inputs to the factory ECU i.e the gas pedal input. Rebuilding our model in this manner created an accuracy of 1.75%.

Furthermore, when we tried to examine the effect of various variables, we discovered that our model had also discovered more efficient and accurate means of arriving to some predictions. We observed that when we changed the engine coolant temperature in our dataset to observe it's effects, our model was using fields that we have not been able to currently decipher for this purpose. The model may have found the oil temperature that it is utilizing to perform it's predictions which is much more accurate. If we had access to that variable we would also utilize it in our predictions. However to enhance the effect of the variables we do know, in this particular case we omitted the variables that we do not know to form our model.

We estimated our data set by using the machine learning model that we mentioned in the previous section created by taking the engine coolant temperature information as an input. In this way, we obtain instant fuel consumption information according to the inputs available. We tried to estimate instant fuel consumption information with

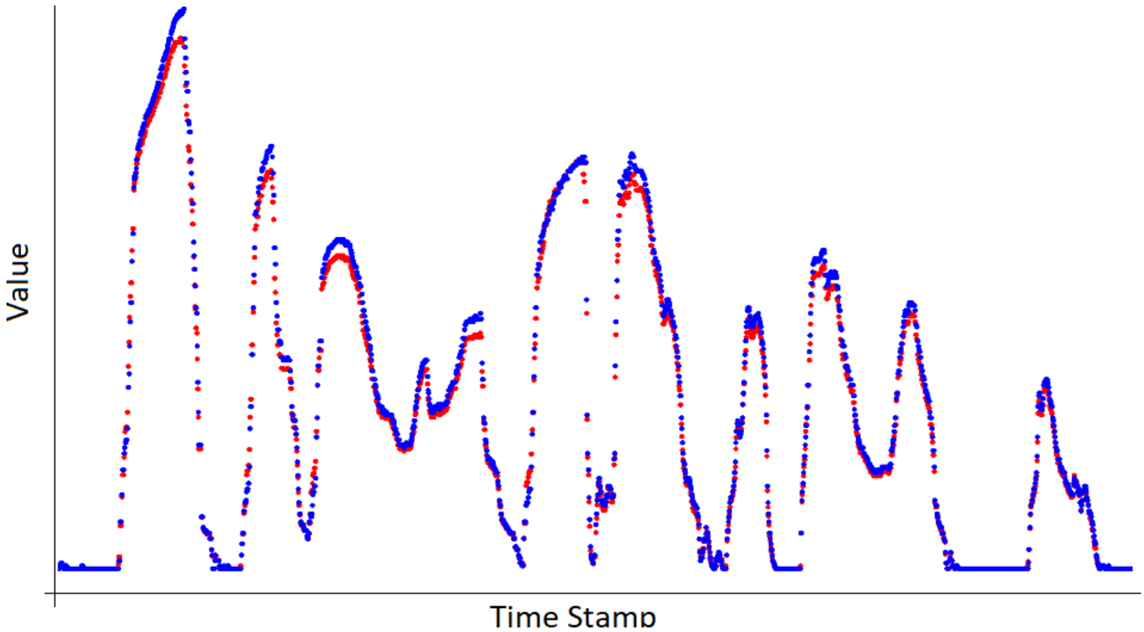

Figure 5.1: Target vs Predictions With All Inputs

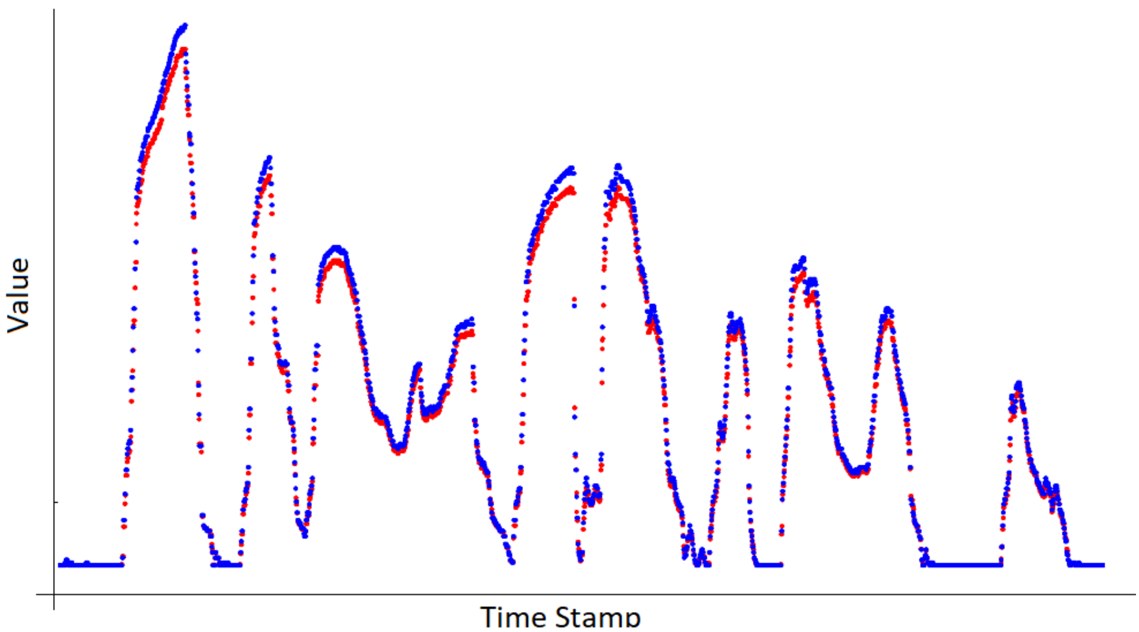

Figure 5.2: Target vs Predictions Without Gas Related Inputs

our pre-trained machine learning model. This model yielded an average of 2.12% error, calculated for 350K predictions between the target and predicted values.

By changing the engine cooling temperature inputs in our data to simulate real conditions. In this way, we determined that instant fuel consumption information varies under various situations according to the engine cooling temperature value. The results obtained are given in the table 5.1. According to these results, if we started the engine with an efficient temperature value at the beginning of the journey, we would be able to save about 5% fuel between 20 and 37° C of engine temperature. When we look at the whole trip (More then 40 minute trip), there is an improvement of approximately 1.3% fuel, but if we consider that the vehicles are not used for a long time in the urban environments, and diesel engines do not heat up quickly, this gain can be considered as an effective result. In addition, it should be noted that the fuel consumption performance of the Chevrolet Aveo 1.3 Diesel vehicle is already at the lower end of the spectrum for vehicles currently sold. Better results can be obtained on vehicles with larger engine sizes. This information is valuable if one is interested in purchasing an engine pre-heating system, either utilizing wall plug electricity or utilizing fuel from the vehicles tank.

Literature regarding engine temperature and fuel consumption show results that corroborate the accuracy of our model According the research of (Bent et al., 2013), it has been determined that, as a result of the improvements made to the engine insulation, the vehicle heats up faster and maintains the engine temperature higher, resulting in 2% savings in fuel consumption. The experiments were performed with 30 minute driving sequences starting with a cold engine. In (Cipollone et al., 2015) , they investigated the effect of heating engine oil faster and calculated the difference by using an electric heater and utilizing exhaust gas directed to the engine. Average fuel savings of 2.8 percent were achieved as a result of their research. A similar study of (Will & Boretti, 2011) on oil heating, measured fuel savings of up to 7% in environments with generally harsh winter conditions. Similar to the above scenarios but different starting conditions our model predicts a difference of 1.3% regarding fuel consumed during a trip (starting from 20 ° C), this demonstrates that our model would have been capable of predicting the results of the above studies without performing additional experiments.

Range	Effect
20 - 37° C	% 5
20 - 50° C	% 3.2
20 - 70° C	% 1.7
20 - 85° C	% 1.3

Table 5.1: Results of fuel gain in different temperatures

6 VALIDATION

This sections aims to validate the model by comparing similar data collected under two different modes with only change in a single variable. The process, although appearing to be trivial at first glance is proving to be anything but. In real driving conditions it is almost impossible to collect data with only a single variable change. Vent though the route chosen was identical, lots of variables come in to play such as traffic, air temperature, wind different acceleration patters. Hence it was the aim of this study to create a model of the vehicles fuel consumption to be able to take all of the variables and produce accurate results.

For validation we chose to investigate the effect of engine coolant temperature on fuel consumption. 2 runs were performed on the same test circuti, one with a pre-warmed engine to 80 C and the second, with an engine starting at 30 C and ending at 80 C at the end of the run. Very early morning hours before traffic were chosen to minimize the effect of trafic, and similar driving style and acceleration patterns were adopted for the data. (Figure 6.1, Figure 6.2)

The runs were approximitely 12 minutes each. For the comparison we present a section with constant speed. It may be observed that there is a difference between the two datasets with regard to fuel consumption. One may also observe that the data with the colder engine consumes more fuel. (Figure 6.3)

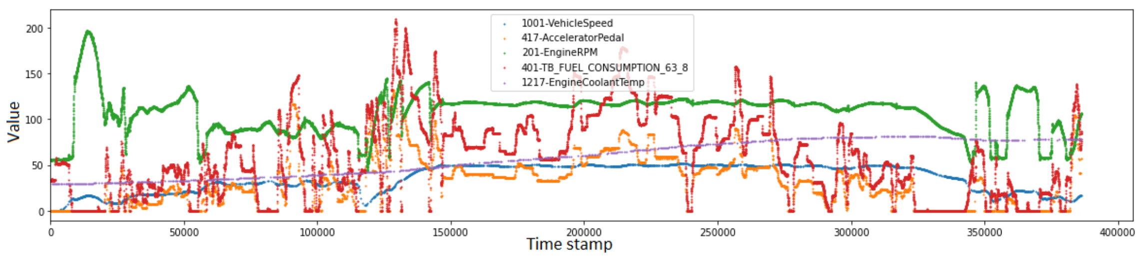

Figure 6.1: Collected Data With Cold Engine

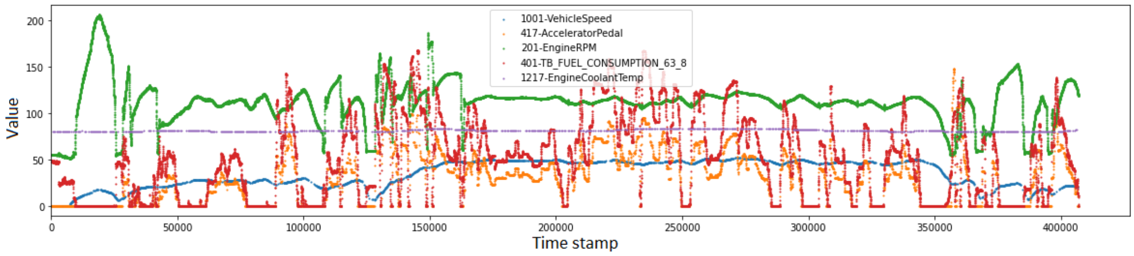

Figure 6.2: Collected Data With Heated Engine

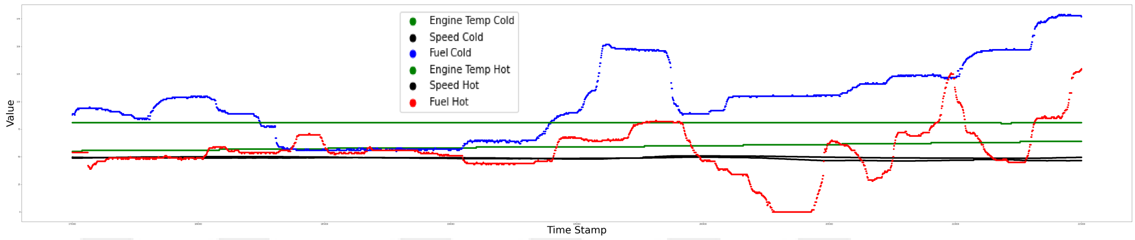

Figure 6.3: Heated Vs Cold Engine Results

To validate our results, we set the engine temperature value of colder engine as 80. Then we estimate fuel consumption and compare with the pre-warmed engine fuel consumption value. The relation between engine temperature and fuel consmption is non-linear, there fore we chose to segment the data to 30-50 C and 30-80 C intervals. The results may be observed in table 6.1

Our model predicted that the fuel consumption would be higher by 7%, and the real world data corraborates this result with a finding of 12% in the 30-50 C region.

Range	Prediction	Validation
30 - 50° C	%7	%12
30 - 80° C	%4	%9

Table 6.1: Validation of Fuel Gain in Different Temperatures

7 CONCLUSION

This study aims to provide means for carbon emission and pollution reduction for a more efficient and cleaner environment by providing the opportunity to analyze methods that can affect and improve instant fuel consumption with minimum cost. For this purpose, a data collection mechanism was designed utilizing a low cost hardware which provides access to a vehicle's Can Bus data. An extensive data set was formed that contains data from all local seasons and 3 distinct drivers via a 2012 Model Chevrolet Aveo 1.3 equipped with a turbo diesel engine. The signals from the Can Bus consisting of unknown data were discovered using machine learning algorithms and domain knowledge. After obtaining meaningful signals, we developed a system that automatically detects inputs that may have an impact on fuel consumption and produces predictive results. We created a binary search tree based model. Due to being less prone to over-fitting and its ability to transition out of local optima, we chose the Xgboost among other rival algorithms. The created system is capable of providing accurate predictions which has 94% R Squared value, at the data-rate of the Can Bus, therefore, each Can Bus signal may be utilized to create an action in real-time.

Although the quality of the data collected and the methods applied are important for machine learning, domain information is also of great importance. The models were augmented with tools to render them highly interpretable in order to facilitate domain expert cooperation. We created a comprehensive system to predict the vehicles fuel consumption with an average error of 1.2%. Our system enables the testing of many different methods to reduce fuel consumption. By injecting scenario inputs to the model, it is possible to analyze how much the vehicles state changes after the related effect has been made. Our system is a very cost effective and fast method to analyze potential modifications to a vehicle. As an example to the prediction capabilities of our system, it has been determined that for an engine coolant temperature of 20 °C to 37 °C, the vehicle consumes %5 more fuel for the same performance output. From this it can be concluded that heating the engine to °C will save %5 of the fuel utilized during the heating phase of the engine, and in the time it takes to reach 50 °C, the engine consumes up to % 3.2 more fuel. If we consider that the diesel engines are warming up late and not able to heat up to the efficient temperatures during general city usage, this improvement in the vehicle can provide us with a well quantifiable fuel consumption improvement. Our analysis reveals that it is feasible to invest in equipment to accelerate the heating phase of the engine. Our system permits the analysis of vehicle fuel consumption with great accuracy for various parameters. Due to its real-time speed, it can also be utilized as the basis for on board reinforcement learning algorithms enabling comparison of the agents actions and their effect on fuel consumption. A database of the resolved Can Bus information has been compiled and contributed to the literature.

Part of this work was presented orally at the III. International Conference on Data Science and Applications 2020. (Naskali & Sen, 2020) and part of the relevant study will be published in the December 2020 issue of the Data Science journal.

8 FUTURE WORKS

Potential directions for future studies are reverse engineering more features, extending model performances, and measurement of the obtained results in real usage data. Since diversification of inputs is crucial for increasing model performance and producing more stable, reliable models, more inputs can be varied with similar reverse engineering methods. It is aimed to obtain more reliable results by analyzing the models we have created in this study from real data by trying mechanical improvements in the vehicle to increase the engine temperature to the efficient range. The data collection system can be augmented with the addition of exhaust emissions hardware to optimize emissions along with fuel consumption.

REFERENCES

(2020). *Eli5 Overview.*
URL https://eli5.readthedocs.io/en/latest/overview.html

(2020). *Joblib Api Documentation.*
URL https://joblib.readthedocs.io/en/latest/

(2020). *Lime Api Documentation.*
URL https://lime-ml.readthedocs.io/en/latest/

(2020). *Machine Learning Challenge Winning Solutions.*
URL https://github.com/dmlc/xgboost/tree/master/demo#machine-learning-challenge-winning-solutions

Bent, E., Shayler, P., La Rocca, A., & Rouaud, C. (2013). *The effectiveness of stop-start and thermal management measures to improve fuel economy*, (pp. 27–39).

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *CoRR*, *abs/1603.02754*.
URL http://arxiv.org/abs/1603.02754

Cipollone, R., Di Battista, D., & Mauriello, M. (2015). Effects of oil warm up acceleration on the fuel consumption of reciprocating internal combustion engines. *Energy Procedia*, *82*, 1–8.

Felix Gers, S. F., Fred Cummins (2015). *Understanding LSTM Networks.*
URL http://https://colah.github.io/posts/2015-08-Understanding-LSTMs/#fnref1

Fugiglando, U., Massaro, E., Santi, P., Milardo, S., Abida, K., Stahlmann, R., Netter, F., & Ratti, C. (2019). Driving behavior analysis through can bus data in an uncontrolled environment. *IEEE Transactions on Intelligent Transportation Systems*, *20*(2), 737–748.

Fuglestvedt, J., Berntsen, T., Myhre, G., Rypdal, K., & Skeie, R. B. (2008). Climate forcing from the transport sectors. *Proceedings of the National Academy*

of Sciences, *105*(2), 454–458.
URL https://www.pnas.org/content/105/2/454

Grossman, R. L., Bailey, S., Ramu, A., Malhi, B., Hallstrom, P., Pulleyn, I., & Qin, X. (1999). The management and mining of multiple predictive models using the predictive modeling markup language. *Inf. Softw. Technol.*, *41*, 589–595.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Comput.*, *9*(8), 1735–1780.
URL http://dx.doi.org/10.1162/neco.1997.9.8.1735

Huybrechts, T., Vanommeslaeghe, Y., Blontrock, D., Van Barel, G., & Hellinckx, P. (2018). Automatic reverse engineering of can bus data using machine learning techniques. (pp. 751–761).

Jakobsen, K., C. H. Mouritsen, S., & Torp, K. (2013). Evaluating eco-driving advice using gps/canbus data. (pp. 44–53).

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S.-I. (2020). From local explanations to global understanding with explainable ai for trees. *Nature Machine Intelligence*, *2*(1), 2522–5839.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.) *Advances in Neural Information Processing Systems 30*, (pp. 4765–4774). Curran Associates, Inc.
URL http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-pdf

Lundberg, S. M., Nair, B., Vavilala, M. S., Horibe, M., Eisses, M. J., Adams, T., Liston, D. E., Low, D. K.-W., Newman, S.-F., Kim, J., et al. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, *2*(10), 749.

Naskali, A., & Sen, B. (2020). Predicting fuel consumption by machine learning with reverse engineering on can-bus data. *3rd International Conference on Data Science and Applications*, *2*, 49–50.

Nasser, S. H., Weibermel, V., & Wiek, J. (1998). Computer simulation of vehicle's performance and fuel consumption under steady and dynamic driving conditions.

In *SAE Technical Paper*. SAE International.
URL https://doi.org/10.4271/981089

Perrotta, F., Parry, T., & Neves, L. C. (2017). Application of machine learning for fuel consumption modelling of trucks. In *2017 IEEE International Conference on Big Data (Big Data)*, (pp. 3810–3815).

Pheanis, D., & Tenney, J. (2003). Vehicle-bus interface with gmlan for data collection. (pp. 88–92).

Remeli, M., Lestyan, S., Ács, G., & Biczók, G. (2019). Automatic driver identification from in-vehicle network logs.

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should I trust you?": Explaining the predictions of any classifier. *CoRR*, *abs/1602.04938*.
URL http://arxiv.org/abs/1602.04938

Sandberg, T. (2001). Heavy truck modeling for fuel consumption simulations and measurements,master's thesis, linköping university.

Szalay, Z., Kanya, Z., Lengyel, L., Ekler, P., Ujj, T., Balogh, T., & Charaf, H. (2015). Ict in road vehicles — reliable vehicle sensor information from obd versus can. (pp. 469–476).

Will, F., & Boretti, A. (2011). A new method to warm up lubricating oil to improve the fuel efficiency during cold start. *SAE International Journal of Engines*, *4*(1), 175–187.
URL http://www.jstor.org/stable/26278144

Zeng, W., Miwa, T., Wakita, Y., & Morikawa, T. (2015). Exploring trip fuel consumption by machine learning from gps and can bus data. *Journal of the Eastern Asia Society for Transportation Studies*, *11*, 906–921.

BIOGRAPHICAL SKETCH

Buğra Şen was born on July 23, 1991 in Malatya. After graduating from Malatya Kolukısa Anatolian High School in 2009, he began to study in Computer Engineering department in Yıldız Technical University. He graduated in 2015 and after that he enrolled in M.Sc. program in Computer Engineering department in Galatasaray University.

Outside of academics, He has been working as a data scientist for the last 3 years at a Garanti Technology Company.

He published a conference paper for M.Sc. thesis, entitlet as "Estimating fuel consumption by machine learning and improving fuel consumption" supervised by Assoc. Prof. Dr. Ahmet Teoman Naskali was presented at "International Conference on Data Science and Applications (ICONDATA), 2020, Turkey" conference.