VERİ BİLİMİ DERGİSİ
www.dergipark.gov.tr/veri

Estimating Instant Fuel Consumption by Machine Learning and Improving Fuel Consumption

A. Teoman NASKALİ[1]*, Buğra ŞEN[2]

[1]Galatasaray University, Institute of Science and Engineering, Computer Engineering, Istanbul
[2]Galatasaray University, Institute of Science and Engineering, Computer Engineering, Istanbul

Makale Bilgisi

Başvuru:
16/07/2020
Kabul:
11/10/2020

Abstract

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 better understood how the variables obtained from the sensors affect the instantaneous fuel consumption and it is proposed to reduce the fuel consumpyeni on between 1% and 2% by interfering with the intake air temperature information. 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.

Keywords: *CAN buss, Machine Learning, Reverse Engineering, Predict Fuel Consumption*[1]

[1] Corresponding Author e-mail: tnaskali@gsu.edu.tr

1 Introduction

The carbon emissions from the transport sector account for a large amount of total carbon emissions in the world. Various methods are being tried to reduce this situation which has an effect on fuel consumption. Examples of such methods are water injection and aero dynamics improvement etc. Such practices can contribute to the economy by reducing carbon emissions as well as consuming less fuel. However, comparing different methods for a 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. The general problem description is to examine a large number of attributes describing a fuel consumption situation and to deploy machine learning methods to find a relation from all the factors for a fuel consumption. It is aimed to understand the results of the applied methods with minimum effort.

Nowadays, Modern cars are very technologically advanced and rely on sensors and actuators which communicate with control units. All the modules communicate each other with using CAN (Controller Area Network) buses which is a central network system in automobiles and other technological vehicles, connects all modules working in the vehicle to the system and thus the components in the vehicle work together efficiently and effectively. Thanks to the can bus protocol, it is possible to understand what is happening in the engine at that moment and 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.

Although every manufacturer uses the same protocol, they do not have to provide the data in the CAN bus exactly the same. For this reason, the information on the bus varies from manufacturer to manufacturer. Reverse engineering is required to make sense of the data on CAN bus. In this work, Most of the data encoded by the manufacturer was determined by reverse engineering and it was found in which bytes the information such as instantaneous fuel consumption and outside temperature were kept. In this way, the necessary steps to reduce fuel consumption have been completed. [1].

2 Related Works

In the paper of Huybrechts & Thomas, They recommend automating reverse engineering analytical measures with a view to enhancing and promoting the process. Applied two application methods that are mentioned, often a basic math strategy and classification-based machine learning techniques. In the first step, data on the CAN bus has been analyzed using simple statistical metrics. This approach provides them, instead of examining thousands of bytes, they were able to reduce their search space. In addition, they estimate information such as speed and speed by comparing the OBD data they collected simultaneously with the Can bus data. They stated that although it is not possible to address all data, it is possible to determine some information on the data path using classification methods.

In order to test the results found in arithmetic methods, visual validation was proposed and the data were addressed in this way. In addition, they recommed that the methods addressed in the study should be tested with more data and data bus information of different manufacturers. There is no general approach for the automated identfy of CAN Ids.[2]

In the paper of Fugiglando, a dataset containing 2135 hours of driving data has been collected for each of the 2418 sensors. The data collection phase took place in 2014 with a total of 55 days of experiment. Cars had been picked up by the truckers in the local office and had to be brought back by the next day. To make it simple, a session-based framework has been built for each user who drives the vehicles. Sampling is not standardized due to the different characteristics of the CAN bus. They only identified signals in a certain frequency range, and extracted information that does not change much, such as seat belt status or outside temperature. For simplicity, they have finalized and selected less than 10 different features. The most importants are gas pedal position, accelerations, vehicle speed, and steering wheel momentum. After extract some features, They used clustering algorithms and get %99 impairing clustering performance.[3]

3 Methodology

The methodology consists of two main parts. The first part includes the data collection phase, and the second part includes modeling steps after analyzing the data which has necessary reverse engineering phases.

3.1 Data Collection

At the beginning of the data collection, We did not proceed with the OBD2 protocol, because the data collection rate was not sufficient to accurately estimate the instant fuel consumption and we could not get the our target variable which name was instant fuel consumption for Chevrolet Aveo 1.3 Diesel (2012). So we focused on to reach CAN bus data directly. To achieve our goal, we used a specific device which is called CANable. The CANable is a lightweight, low-cost, open-source USB to CAN adapter that appears as a virtual serial port on your device and functions as a serial line to the CAN bus interface. It also has python library support for data logging operations.

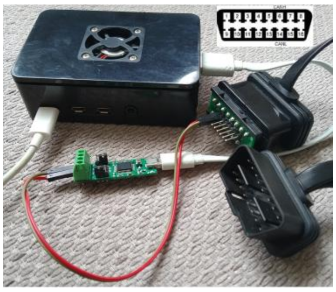

Figure 1. Can Bus Logging System

We designed a basic python flow for logging process and run this code in a Raspberry 4 sigle board computer which has directly connected car's CAN bus line with CANable device. The device which feed it from the vehicle's 12v input. Thus, the vehicle has been continuously collecting data from the moment it starts working until it stops. More than one hundred trip data was collected by two different users over a 6 months period which starts in October. You can see the details of the logged data below.

3.2 Finding an Entry Point

As there are 66 different message ID's on our CAN BUS, 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. Instead the approach adopted in this work is to find the fuel consumption information and utilise 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 the receive. The canbus signal that first appears on the canbus also had the lowest ID, indicating that it has a higher priority.

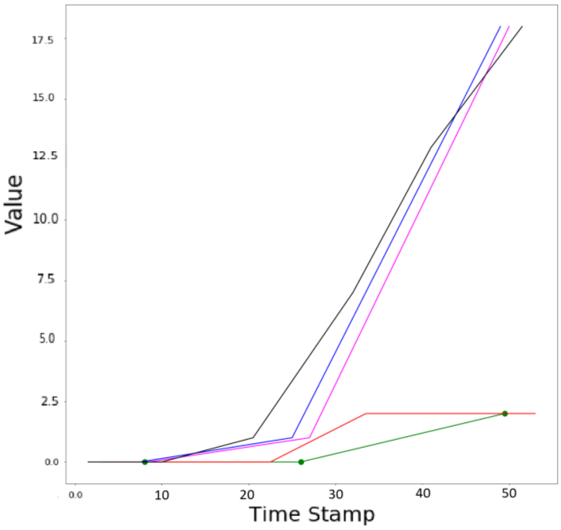

Figure 2. To Understand signal order

3.3 Obtaining Mass Air Pressure, Mass Air Flow and Intake AirTemperature Information

It was important to determine which ids contain information such as intake air temperature, mass air flow and mass air pressure sensor values. This values have a great importance and direct effects on fuel consumption. If we want a establish an understandable and accurate prediction model, we must detect this values. To develop a prediction model, we must know and to be sure this information To design a predictive model.

To achieve this aim, we collected five different data which has 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 reaching locations disassemble instructions, we collected data by removing the Maf sensor of the vehicle and blowing air to the sensor sequentially with a hair dryer and leaf blowing machine. This process was repeated with the vehicle engine not running and running. n this way, we tried to determine what information could be in which id by drawing the data we collected.

To identify Map signal, we collect similar data from vehicle but this time we disassemble vehicle's air flow pipe of the engine and fixed the leaf blowing machine on this pipe with the help of isolation tape and manually blow air through this pipe to the engine. This experiment was repeated with the engine running and not running which is like the previous one. 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 plot all the signals in our database. Than We collected all signals that they have any 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 which signal contains IAT information. When we look into engine running statement. We can see the negative pressure on the collected graph. Such a situation is not observed when the engine is not running and this makes sense to us. Also we can detect number of blowing count with the leaf blowing machine on the chart.

3.4 Determining the Fuel Consumption Field

To find out target variable, we developed a machine learning model which has trying to predict gas pedal information. As a expert opinion, fuel consumption and gas pedal strongly correlated to each other. After build machine learning model, we collect feature importances of the all variables and remove all the non related features one by one with the analyse their plots.

4 Methodology

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 tempereature and lots of other factors enter 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 some data preprocessing steps. First of all, Line-based data consisting of message ID's are transformed into signal-based variables by pandas pivot operation based on index value as an group by option. At this stage, since the frequency of sending some ids in can bus is different from each other, there will be no data corresponding to each index value grouped, so the data starts to be missing. We filled this missing data with the last data we could read by pandas fillna fucntion with ffill parameter. 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 which applied pivot operation.

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 there can be more than one message ID's that indicates the fual consumption what we can not detect in reverse engineering step. If we don't realize this situations, we wouldn't have done a logical prediction model. To overcome this problem, we experimented with input combinations iteratively, examined the variables that directly explain the model with a highly 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 we are planning to implement real-time forecasting on raspberry pi, so the estimates we will produce must be at a acceptable speed. Therefore, it is a problem to process all inputs and generate estimation value on raspberry with lower processing capacity. As a solution to this problem, we proceeded by finalizing our model with the first 20 variables, taking into based on the order of importance of the variables describing the 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 improvements in training time, We were able to spend more time on parameter optimizations and we also produced more interpretable models by selecting less inputs.

5 Methodology

Due to the black box algorithm (Xgboost) we applied and many inputs that we do not know their possible effect to our target variable, 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. Beause of the local and global explanatory capabilities, Shap was the package we preferred to progress. In this way, we were able to deduce and interpret how the inputs entered into the model and affect the predicted values.[4]

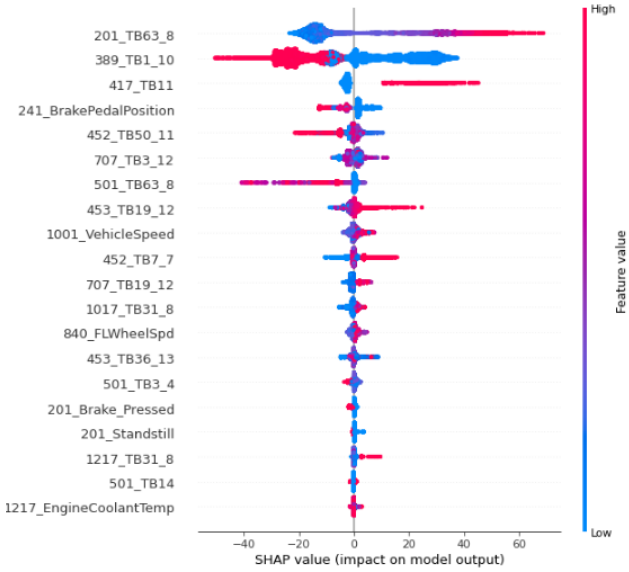

Figure 3. Feature Importances of Prediction Model

When the model generating estimates, it is seen in which intervals the inputs produce high and low estimates. Red dots are high estimates and blue dots are low estimates. During the analysis the variables that entered 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 example of the reason, 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 tries to separate this information in the top nodes. [5]

It is also possible to visualize what kind of estimates the variables we use as input in the model produce for all estimates produced. We can summarize this as global explanation. Figure 4 you can see the global explanation of the model for engine cooling temperature inputs.[6]

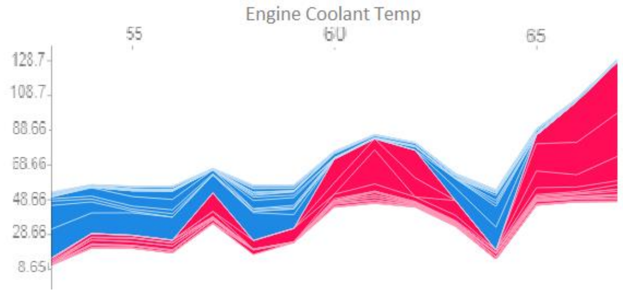

Figure 4. Global Effect of Engine Cooling Temp

If we examine local explanatory examples in Figure 5, we can see how much input has 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.[7]

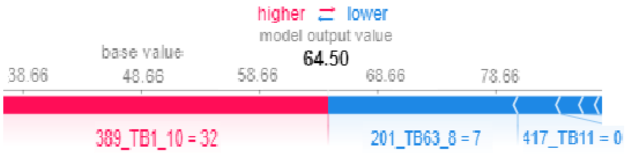

Figure 5. Single Prediction Analysis (Local explanation)

Table 1. Intake Air Temperature Effect for Increase Fuel Consumption

Intake Air Temp	Vehicle Speed	Effect
10 °C	0-40 km	% 1.17
10 °C	>40 km	% 2.1

6 Investigating the Effect of Intake Air Temperature on Fuel Consumption

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.

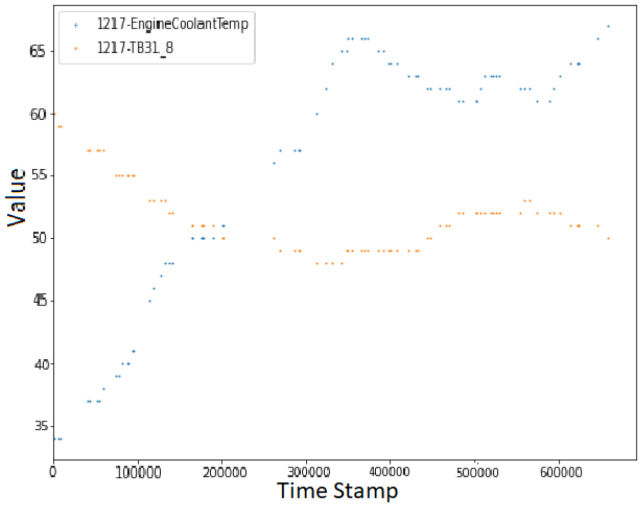

Figure 6. IAT Journey Example Informations

In Figure 6, orange points represent our IAT values without -40 offset value. So, the value of 50 appears on the graph actually corresponds to 10 degrees Celsius.

7 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. In the previous sections, we explained how we overcome this problem with limited computing and memory power. We use the 4gm memory version of Raspberry pi 4 and this configuration is sufficient to generate a scores in milliseconds time interval.

We build our model which has 1.0.2 version of xgboost algorithm. After installing the same version on raspberry, we transferred the scoring object to raspberry thanks to the python joblib library. This library allows us to save and read objects in a binary format. Model objects holds some informations such as feature names and model parameters. We use that information and design a adaptive scoring process. This allows us, If we generate new model, we do not need to update our scoring code. We made this structure in order not to waste effort if we want to train the model real time. When we did the scoring tests, there was no difference in the scores. For this reason, there is no need for other methods such as Pmml.

8 Results and Conculution

This study aims to reduce the carbon emission and obtain 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, data collection mechanism was designed with raspberry pi 4 and canable device which can read on vehicle's Can Bus data. Data was collected in approximately 10 months with the help of 3 different people via 2012 Model Chevrolet Aveo 1.3 With a diesel vehicle.

The hardest part of this work was reverse engineering steps which has solved with several methods, such as data visualization, statistical methods and machine learning approaches. After the signals read from the Can bus line are labeled, various machine learning models have been set up that estimates instant fuel consumption. Although the quality of the data collected and the methods applied are important for machine learning, domain information is of great importance. For this reason, we aimed to increase the legibility of the models established using the shap library which is developed in python language.

Many different methods can be tried to reduce fuel consumption. By changing the inputs of the model, which was established with the machine learning we developed with this study, it was possible to analyze how much gain can be made after the related change has been made, and to be able to learn it with minimum time and money loss within the scope of benefit cost.

With the machine learning approach we have developed, it has been determined that in order to decrease Intake air temperature from 30 degrees celsius to 20 degrees celsius, We conducted 2 experiments, above and below 40 km/h speed. it will be possible to save %1.17 and %2.1 of benefit of fuel consumption. If we consider that the decreasing 10°C temperatıre of intake air temperatıre on the vehicle by spraying water into the carburetor, this improvement in the vehicle can provide us with a fuel consumption. With this method used in some vehicles today, it seems like a feasible method and With such small interventions, we can have vehicles with more environmentally friendly carbon emissions.

9 Future Works

Potential directions for future studies are reverse engineering more features, extends 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.

References

[1] Pheanis, David & Tenney, Jeffrey. (2003). Vehicle-Bus Interface with GMLAN for Data Collection.. 88-92.

[2] Huybrechts, Thomas , Vanommeslaeghe, Yon , Blontrock, Dries , Van Barel, Gregory , Hellinckx, Peter. (2018). Automatic Reverse Engineering of CAN Bus Data Using Machine Learning Techniques. 751-761. 10.1007/978-3-319-69835-971.

[3] U. Fugiglando et al., Driving Behavior Analysis through CAN Bus Data in an Uncontrolled Environment in IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 2, pp. 737-748, Feb. 2019.

[4] 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 toglobal understanding with explainable ai for trees.Nature Machine Intelligence,2(1), 2522–5839.

[5] Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting modelpredictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus,S. Vishwanathan, & R. Garnett (Eds.)Advances in Neural Information ProcessingSystems 30, (pp. 4765–4774). Curran Associates, Inc.

[6] 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). Explainablemachine-learning predictions for the prevention of hypoxaemia during surgery.Nature Biomedical Engineering,2(10), 749.

[7] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). ”why should I trust you?”:Explaining the predictions of any classifier.CoRR,abs/1602.04938.