A NOVEL APPROACH FOR OBSERVATION OF MODIFICATION’S IMPACT ON VEHICLES FOR FUEL CONSUMPTION THROUGH MACHINE LEARNING

(ARAÇ ÜZERİNDE YAPILAN MODİFİKASYONLARIN YAKIT TÜKETİMİNE OLAN ETKİSİNİN MAKİNE ÖĞRENMESİ MODELİ KULLANILARAK GÖZLEMLENMESİ İÇİN YENİ BİR YAKLAŞIM)

by

Cem YENİÇERİ, 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

June 2019

This is to certify that the thesis entitled

A NOVEL APPROACH FOR OBSERVATION OF MODIFICATION'S IMPACT ON VEHICLES FOR FUEL CONSUMPTION THROUGH MACHINE LEARNING

prepared by **Cem YENİÇERİ** in partial fulfillment of the requirements for the degree of **Master of Science in Computer Engineering** at the **Galatasaray University** is approved by the

Examining Committee:

Assist. Prof. Dr. Ahmet Teoman NASKALİ (Supervisor)
Department of Computer Engineering
Galatasaray University ------------------------

Assoc. Prof. Dr. Özlem Durmaz İNCEL
Department of Computer Engineering
Galatasaray University ------------------------

Assist. Prof. Dr. Ayşegül ERMAN
Department of Computer Engineering
Işık University ------------------------

ACKNOWLEDGEMENTS

I would like to acknowledge and thanks to the following important people who have supported me, not only during the course of this project but throughout my master’s degree. Without their support, it is not easy for me to finish my graduate education uneventfully.

First of all, I would like to express my deep gratitude to my master thesis advisor, Assist. Prof. Dr. Ahmet Teoman Naskali for his continuous support, motivation and knowledge during my master study. He gave considerable support to me for planning and performing my master’s degree.

I am also grateful to my family members which include my parents and my sisters for their moral support. They always encouraged me through my master’s degree since the very beginning.

Last but not the least important, I appreciate that my close friends, Tolga Tuna and Dilbeste Saglam, always help for providing valuable suggestions about this study.

And I'm also grateful to Bekir Osmanpasaoglu for sharing his wisdom regarding VCDS.

Cem YENİÇERİ

TABLE OF CONTENTS

LIST OF SYMBOLS

ABS	: Anti-lock Braking System
AI	: Artificial Intelligence
AIS	: Automatic Idle Speed
ANN	: Artificial Neural Networks
API	: Application Programming Interface
ASR	: Anti-Slip Regulation
CAN	: Controller Area Network
CFD	: Computational Fluid Dynamics
CPU	: Central Processing Unit
DBW	: Drive-By-Wire
DLC	: Data Link Connector
DTC	: Diagnostic Trouble Code
ECU	: Engine Control Unit
EDS	: Electronic Differential System
ESP	: Electronic Stability Program
FCM	: Fault Code Memory
GBM	: Gradient Boosting Machine
GBRT	: Gradient Boosted Regression Tree
GPS	: Global Positioning System
GPU	: Graphics Processing Unit
IAC	: Idle Air Control
ISC	: Idle Speed Control
LPG	: Liquefied Petroleum Gas
ML	: Machine Learning
OBD	: On-Board Diagnostic
OEM	: Original Equipment Manufacturer
OS	: Operating System
OSI	: Open System Interconnection
PWM	: Pulse Width Modulation
RPM	: Revolutions Per Minute
TCS	: Transaction Control System
TSI	: Turbocharged Stratified Injection
UDS	: Unified Diagnostic Services
VAG	: Volkswagen Audi Group
VCDS	: Windows-based Diagnostic Software
VPW	: Variable Pulse Width

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

ABSTRACT

Vehicles have become indispensable in modern societies. Although they provide great convenience, they also come with a lot of serious problems. Especially, vehicles with high fuel consumption and low efficiency have negative impacts on the world and vehicle users. Vehicles exhaust harmful gases by burning fuel, which affects the environment by causing air pollution. High fuel consumption also affects the driver's financial situations in a negative manner.

Most cities do not have the electrical infrastructure to fully support the electrification of all vehicles indicating that internal combustion engines will still be utilized for several decades.

Vehicle producers continue to investigate, and research means for decreasing fuel consumption and increasing efficiency of internal combustion engines but observing effects of any recent modifications on a vehicle requires expensive and long-term test processes. The main aim of this study is decreasing the required duration to a minimum and providing an easy comparison with the state of the vehicle before modifications.

In this thesis, we investigate various sensor data from vehicles and identified those, which have an effect on fuel consumption. The fuel consumption model of a vehicle is formulized via the data collected from vehicle sensors by taking environmental factors into consideration using various machine learning algorithms. Artificial Neural Network, Gradient Boosted Trees and XGBoost (eXtreme Gradient Boosting) algorithms are utilized and their results are compared. According to the results, even the performances are observed to be close to each other, XGBoost algorithm provided the best results since training data is not large enough for deep learning at this stage. After an appropriate model is obtained, impacts on the fuel consumption of modifications are observed compared to the previous state of the vehicle. In this way, expensive and long-term test processes are not required to identify the contribution of modifications.

Keywords: Vehicle Fuel Consumption, Fuel Economy, Machine Learning, CAN Bus

ÖZET

Araçlar günümüzde toplumun vazgeçilmez bir parçası haline geldi. Her ne kadar hayatımıza büyük kolaylıklar sağlasalar da ciddi sorunları da beraberinde getirmektedirler. Özellikle yüksek yakıt tüketimine ve düşük verimliliğe sahip araçlar hem içinde yaşadığımız dünyayı hem de kullanıcısını birçok yönden olumsuz etkilemektedirler. Araçların yakıtı yakarak dumanlarını doğaya salıyor olmaları atmosferi ve ozon tabakasını etkilemekte ve yoğun hava kirliliğine sebep olmaktadır. Kullanıcı açısından baktığımızda ise aracın yakıt tüketiminin yüksek olması ve petrol fiyatlarının giderek yükselmesi kullanıcısını maddi açıdan olumsuz etkilemektedir.

Çoğu şehirlerde hala tüm araçların elektrikli olmasına elverişli olacak bir altyapı henüz olmadığından içten yanmalı motorlar uzun bir süre daha kullanılmaya devam edecektir.

Araç üreticileri yakıt tüketimini düşürmek ve verimliliği arttırmak için yatırımlar ve çalışmalar yapmaya devam etmektedirler ancak araç üzerinde yapılan bir modifikasyonun etkisini gözlemlemek için aracı uzun ve masraflı test süreçlerine tabi tutmaları gerekmektedir. Bu çalışmanın en büyük amacı bu süreci minimuma indirerek araç üzerinde yapılan bir modifikasyonun, aracın bir önceki yakıt tüketim değerleri ile kolayca karşılaştırılabilir olmasını sağlamaktır.

Bu çalışmada literatür araştırması ile araç üzerinden toplanabilecek sensor verileri belirlenmiş ve yakıt tüketimine etkisi olan veriler tespit edilmiştir. Aracın yakıt tüketimi, çeşitli makine öğrenmesi yöntemleri aracılığıyla çevresel etkenler de göz önünde bulundurularak araç üzerindeki sensor verileri ile formülüze edilmiştir. Modeller Artificial Neural Network, Gradient Boosted Trees ve XGBoost (eXtreme Gradient Boosting) algoritmaları kullanılarak eğitilip, sonuçları karşılaştırılmıştır. Her ne kadar tüm sonuçlar birbirine yakın olsa da, XGboost algoritması en iyi sonucu veren algoritma olmuştur. Çünkü şuana kadar araç üzerinden toplanan very boyutu derin öğrenme algoritmaları için yeterli değildir. En efektif model elde edildikten sonra, araç üzerinde yapılan modifikasyonların yakıt tüketimine olan etkisi, aracın bir önceki durumu ile karşılaştırılmıştır. Böylece modifikasyonların yakıt tüketimine olan etkisi uzun mesafeli ve uzun sureli testlere gerek olmadan tespit edilebilmiştir.

Anahtar Kelimeler: Araç Yakıt Tüketimi, Yakıt Ekonomisi, Makine Öğrenmesi, CAN Bus

1. INTRODUCTION

Nowadays, vehicle fuel efficiency and fuel consumption are the most important topics all over the world (Anup Bandivadekar, 2014). It can negatively affect drivers' economic situations and environmental pollution if consumption increases and efficiency decreases.

Figure 1.1 The Trend in Gasoline Prices in the U.S [1].

In Figure 1.1 shows the gasoline prices trend in the United States from 1990 to 2018. Generally, the trend is always the same across the countries, but prices are different because of countries' different tax policies. According to the gasoline price trend chart, in 2018, one gallon of gasoline is 2.72 U.S. dollars, though it was 1.3 U.S. dollars in 1990.

[1] https://www.statista.com/statistics/204740/retail-price-of-gasoline-in-the-united-states-since-1990/

In 2012, it peaked at 3.62 U.S. dollars and declined in the following years, but it started to increase in 2017 and it still continues nowadays.
As the price of fuel increases, many drivers are investigating ways to reduce a car's fuel consumption. The problem is not only experienced by drivers, but also vehicle manufacturers because of government policies and competitive pressures with other manufacturers.

Manufacturers recommend more efficient driving techniques. One of which is about selecting a higher gear. They also developed an engine start-stop system, which automatically shuts down and restarts the internal combustion engine to reduce the amount of time the engine spends idling. Furthermore, hybrid cars are becoming popular day by day which manufacturers are encouraged to invest heavily. Recently, Volkswagen group has developed active cylinder technology. The main aim of the system is to reduce significantly fuel consumption by temporarily shutting off two of the four cylinders during low to mid loads. According to Volkswagen, cylinder deactivation reduces fuel consumption of the 1.4 TSI by 0.4 litres per 100 km[2].

Beside manufacturers, drivers apply some techniques advised by their engine mechanics or in media to reduce fuel consumption which are using LPG system, reducing unnecessary weights, making their car more aerodynamic, keeping the windows closed, minimizing air conditioning, maintaining tire inflation and changing tires with more appropriate ones or using cruise control.

Currently, drivers and auto manufacturers are not able to precisely measure modifications relating to fuel consumption. In modern cars a display is provided to the user that shows the amount of fuel consumed per unit of distance, although this information is useful as an indicator to the vehicle operator, relative changes in driving behaviour or aerodynamic modifications, engine modifications, fuel additives cannot be observed. To make use of this information, one has to know what the fuel consumption was before. There are very many parameters that affect fuel consumption and no two on-road tests are identical.

[2] https://www.greencarcongress.com/2011/09/zas-20110902.html

Although standard traffic-free routes are used for fuel consumption calculations parameters such as air temperature and wind can affect the results significantly. Currently, engine testing is done in laboratory environment where the environment is strictly controlled. Aerodynamic tests are usually performed in wind tunnels or CFD programs are utilized to calculate the drag.

Our aim in this work is to be able to estimate the fuel consumption of a vehicle using various vehicle sensor data by building a machine learning model. This model will later be used to test the efficiency of modifications.

Luckily, new modern vehicles provide many detailed information about the current state. They have over 100 controllers (ECU). Airbags, cruise control, braking, audio systems, mirror adjustment so on and so forth, these systems can communicate with each other via a network called the CanBus. The CanBus system acts as a central networking system in vehicles like the nervous system in the human body. Once one listens to these communication systems and applies reverse engineering, one can understand how a vehicle behaves at a given time. Obtaining this data provides a big advantage for researchers. Logging all these necessary data enables analysing, monitoring and developing new features.

Accessing CanBus data and decoding it was not straight-forward compared to now. Nowadays, several manufacturers provide API for developers and people who are interested in their own vehicles. For instance, Ford Motor Company announced its open source platform OpenXc which provides a metric from a car’s internal network by only installing a small hardware module to read and translate data. Tesla is also another game changer, provide metrics to its customers over a Restful API. It would appear that most of the other manufacturers will provide similar services in the near future to not keep away from the competition.

Main Research Question:

Is it possible to use various sensor data to model vehicle fuel consumption? Is it possible to compare and observe the effects of any modifications which is made to reduce fuel consumption of the vehicle with the previous original set-up of the vehicle?

In order to find answers to our research questions, we started our research with a literature review. Meanings of the data which can be collected from the vehicle over CAN Bus are researched. The literature review shows that there are certain protocols in new generation vehicles, and it is possible to access many vehicle's data by using these protocols.

After the literature review, we obtained the necessary equipment. We started to collect the required data from our test vehicle, a Volkswagen Passat 1.8 TSI model. We collected a lot of data for long kilometres under certain conditions. With the collected data, several fuel consumption models of the vehicle were established, and their performances are compared. With the most appropriate model, we observed positive or negative effects of any modification on the vehicle for reducing fuel consumption in a less time-consuming, an easy and a cheaper way.

The rest of the thesis is organized as follows: In Chapter 2, we provide a literature overview. In chapter 3, explanation of used devices and methodologies and also details of data which are taken from vehicles are given. In Chapter 4, we explain the followed solution methodology while in Chapter 5, we present the computational studies. In the last chapter, which is Chapter 6, conclusion and future works are discussed.

2. LITERATURE OVERVIEW

In this section, literature research is made about vehicle fuel consumption modelling with machine learning.

There are also several studies to estimate fuel consumption using machine learning for a variety of reasons. (Wickramanayake & Bandara, 2016) built fuel consumption model for fleet management using several machine learning algorithms. Their aim is to prevent fraudulent activities and increase consumption efficiency in fleet management.

In another study, (Rahman & Smith, 2017) predicted fuel consumption for commercial buildings. In their study, Artificial neural network (ANN), Gaussian process (GP) and multivariate linear regression and ridge regression were employed using weather and schedule parameters. They show that ANN and GP perform better than others.

(Du, et al., 2017) described the relationship between fuel consumption and the impact factors, massive amounts of vehicle data. They established the Back Propagation Neural Network model to forecast fuel consumption of the vehicle.

(Çapraz, et al., 2016) aimed to achieve the best prediction method for forecasting fuel consumption for long distance. They used Multiple Linear Regression (MLR), Artificial Neural Network (ANN) and Support Vector Machine (SVM) and consider that SVM model of fuel consumption offers the best result with 0.9 R-value for their study.

(Zeng, et al., 2015) studied large-scale CAN Bus and GPS data to predict fuel consumption. They applied several machine learning algorithms and their study shows that SVM performs better than others.

(Perrotta, et al., 2017) presented a machine learning model for fuel consumption of trucks. Their main aim is to help fleet managers that can enhance fleet vehicle routing decisions. As a result of their study, Random Forest performs better than SVM and ANN.

(Almèr, 2015) described a machine learning algorithm to predict fuel consumption in his master thesis. His main aim is finding the most appropriate machine learning methods for forecasting, showing the effect of data collection frequency on prediction and investigating the most powerful features for fuel consumption.

Unlike the examples above; In this study, in addition to building the fuel consumption model of the vehicle, we tried to observe the effects of our modification by using this model.

We also used ANN in our study, which is one of the Machine Learning algorithms used in the literature. Since we did not have enough data, we found better results using another Machine Learning algorithm, that is XGBoost instead of ANN.

3. BACKGROUND

In this chapter, the experimental setup and engine specification information are given. In addition to these, vehicle communication protocols which are CANBUS, OBD2 and UDS, are explained in detail. Moreover, an introduction to data acquisition is given and definitions for collectable data from the vehicle are investigated.

3.1. Experimental Setup

Tests vehicle specifications are given in the Table 3.1.

Table 3.1 Vehicle Specifications

Brand	Volkswagen Passat
Model Year	2010
Engine Type	1.8 TSI
Fuel Type	Petrol
Transmission	Automatic (DSG)

3.2. Engine Specifications

Engine specifications are given in the Table 3.2.

Table 3.2 Engine Specifications for Passat 1.8TSI

Power	160 hp (118kw)
Torque	250Nm
Consumption	7.4 - 8.2 l/100km
CO2 Emission	180 - 195 g/km

3.3. CANBUS

Controller Area Network (CanBus) is a vehicle communication standard. In an application, all microcontrollers share information with each other without a host computer via CanBus. It is a message-based protocol which is designed for electrical wiring. Although it is used within vehicles to save on copper expense, it is also used in many other contexts (Corrigan, 2016).

As a short history of CanBus, it started in 1983. In this year CanBus showed up. The first version of CanBus protocol is designed by Robert Bosch. In 1986, the protocol was released officially and also it was introduced to society at the Society of Automotive Engineers conference in Michigan in that year. Intel and Philips are the first producers of CanBus controller processors in 1987. In 1991, First CanBus multiplex wiring system is used in Mercedes-Benz W140 in 1991[3].

Several versions of CanBus protocol were published. In 1991, the latest version which is known as the CanBus 2.0 specification was introduced. CanBus 2.0 specification consists of two parts. These parts are Part A and Part B which are for the standard format with an 11-bit identifier and for the extended format with a 29-bit identifier respectively. A CanBus device that uses 11-bit identifiers is called CAN 2.0A and 29-bit identifiers is called CAN 2.0B.

In these days, vehicles have many various of subsystems like transmission, airbags, braking systems (ABS), cruise control, electric power steering, entertainment systems, electric power windows, self-closable doors, electronic mirror adjustment, etc. These subsystems are controlled by electronic control units. Modern vehicles have also the engine control unit (ECU) which is the most powerful processor vehicle has. Electronic control units and engine control unit have to communicate with sensors and get feedback from them. They have to communicate with their actuators. Therefore, the CanBus standards become prominent.

---

[3] https://www.can-cia.org/can-knowledge/can/can-history/

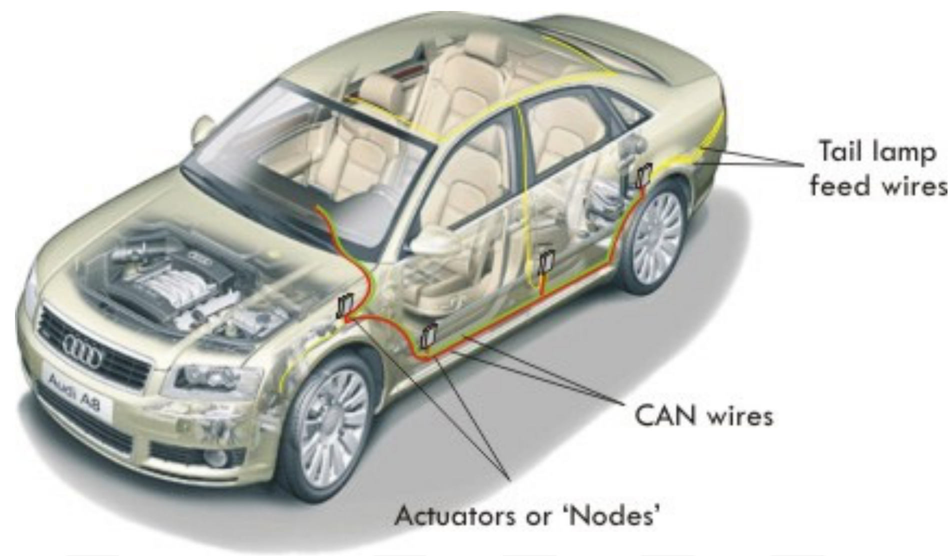

Figure 3.1 CAN Bus Communication[4]

CanBus is a multi-master serial bus communication standard. It provides a connection between electronic control units, sensors and actuators. These connections are called nodes. CAN network contains lots of nodes. The node may also be a gateway which provides the communication between the network and an outer computer (such as a laptop or mobile device) over a USB or Ethernet port to the devices on a CAN network.

In the CanBus network, the messages to be transmitted are prioritized according to their message identifier. Each unit in the network is synchronized to sample every bit on the CAN network at the same time. Each unit has to listen to the CanBus network line. Since each unit must ensure that the network line is free before sending any data to the network. These synchronization and data transfer method called as bitwise arbitration of contention resolution.

3.4. OBD2

Modern vehicles have a standardized system for providing communication between computer systems of vehicles and external devices like personal computers and

[4] https://www.rightconnections.co.uk/canbus/

smartphones. This standardized system called OBD (On-Board Diagnostic) [5]. The system is used for monitoring the vehicle electronic systems and accessing malfunction details on the vehicle (Neriya, et al., 2017).

Before the OBD systems, indicator lights are used to display malfunctions and problems on vehicles. That simple system would not provide detailed information like the occurred time and number of occurrences about the problem. On the other hand, modern OBD systems provide detailed status information for each electronic system of the vehicle (Dude, 2018). Details of malfunctions and problems are obtainable with OBD systems.

First version of OBD (OBD-I) was presented in California in 1991. This version of OBD system focused on monitoring certain essential modules like engine control unit, fuel transfer systems, oxygen sensing systems, and exhaust gas sensors. Diagnostic Trouble Codes (DTCs) were introduced with the first version.

In 1996, the OBD-II standard was introduced. Monitoring capabilities of the OBD-I standard was improved and the number of monitorable vehicle module was increased. With the version II, Europe and the United States manufactured vehicles must have the OBD-II system. It also became obligatory by law in these years. With the OBD-II standard, type of connector, pinouts of a connector, electrical signalling protocols and messaging formats were specified, and list of vehicle monitoring parameters was provided to manufacturers and customers.

There are various applications and devices to get data from the vehicle OBD system. Hand-held scan devices, data logger devices, mobile device-based tools and applications, computer-based scan tools and telematics are known devices and applications. These devices are connected to the vehicle OBD interface with Data Link Connector (DLC). DLC is a 16-pin standard connector. Connector shell in the vehicle is generally located near the drivers, under the dash or the vicinity of the ashtray.

---

[5] https://en.wikipedia.org/wiki/On-board_diagnostics

In this thesis, Konnwei KW902 (Wi-Fi version)[6] model is used as OBD-II adapter. This device is a small adapter which can provide vehicle data via Bluetooth or wi-fi connection to a mobile application. Using the adapter, it is possible to access vehicle data which are fuel consumption, momentum, vehicle speed, engine load, oil pressure, coolant temperature, intake manifold pressure, throttle position, oxygen sensor values and more.

Figure 3.2 Konnwei KW902 (Wi Fi version)

There are five different communication protocols available for the OBD-II specifications. The short explanation for each is given below (Dude, 2018):

SAE J1850 PWM: This signal is Pulse Width Modulation. It runs at 41.6 kbps. This protocol is generally used on Ford brand vehicles.

SAE J1850 VPW: This protocol is Variable Pulse Width. It runs at 10.4 kbps. GM brand vehicles use this protocol.

---

[6] http://www.konnwei.com/

ISO 9141-2: Chrysler, European, or Asian vehicles use this protocol. It runs at 10.4 kbps. It is asynchronous serial communication.

ISO 14230 KWP2000: This is the Keyword Protocol 2000, another asynchronous serial communication method that also runs at up to 10.4 kbps. This is also used on Chrysler, European, or Asian vehicles.

ISO 15765 CAN: This protocol has been mandated in all vehicles sold in the US from 2008 and later. However, if you have a European car from 2003 or later, the vehicle may have CAN bus. It has a the two-wire communication method and runs at up to 1Mbps.

3.5. UDS

Unified Diagnostic Service (UDS) is a protocol which provides communication between the vehicle ECU and diagnostic systems [7]. Any occurred error can be monitored with the UDS protocol. This protocol allows reprogramming an ECU. Any occurred error on the ECU memory is readable and removable via UDS protocol.

UDS protocol is called as unified because it combines several standards like KWP 2000 and diagnostics on CAN (ISO 15765). UDS is a global and not a company-specific standardized protocol. Tier 1 suppliers of Original Equipment Manufacturer (OEM) use this protocol in their newly produced ECUs.

UDS protocol is designed based on the Open System Interconnection (OSI) model. OSI is a model that standardizes the communication functions of the computing system without regard to its technology. OSI model partitions a communication system into layers. Therefore, UDS also has a layered architecture. UDS tools and services are using the fifth and seventh layers of the OSI model.

---

[7] https://www.iso.org/standard/55283.html

Modern vehicles have a diagnostic interface for off-board diagnostics. A diagnostic tool (tester tool) or a host computer (client) can connect to the vehicle bus system through the vehicle diagnostic interface. It is possible to send predefined UDS messages to the vehicle control units over the connection. Interrogation a fault from control unit memory, getting data from a control unit and updating a unit with a new firmware are some functions of UDS has.

List of 4 important categories of services and capabilities provided by a UDS protocol is given below [8]:

Upload/Download Capabilities: UDS protocol supports ECU programming. It is possible to update ECU software with a new version. This is required to resolve any existing bug. Using the upload and download capabilities of UDS protocol, large packets of data can be sent and received to and from the car's ECU for ECU reprogramming purpose.

Remote Routine Activation: Vehicle Diagnostics may require testing the faulty component in a given range of parameters. Moreover, during the testing phase of the vehicle, some system tests may be required to run over a period of time. For all such tasks, remote routine activation service of UDS protocol is used.

Data Transmission Capabilities: The data transmission capabilities enable the connected host device to read and write ECU information. The host device can also read data from the physical memory at the specified address. The information can range from static info like ECU serial number to some real-time data like the current status of the sensors, engine speed etc.

Fault Diagnostics: Fault diagnostics is another important service UDS has. Diagnostic trouble codes (DTCs) are saved to ECU fault code memory (FCM). When an error occurred in the vehicle, a DTC is generated and saved FCM. With a fault, diagnostic service of the UDS can read these saved DTCs from FCM of the ECU. It has access to both emissions related or non-emission related DTC information.

---

[8] https://www.embitel.com/blog/embedded-blog/4-uds-protocol-services-every-automotive-geek-should-know

3.6. Data Acquisition

3.6.1. Data Acquisition Tool (VCDS)

In this study, VCDS [9] is used as a data acquisition tool. It is Windows-based diagnostic software for VW/Audi/Seat/Skoda.

Normally, an OBD2 scan tool could be used for data logging but the VCDS is only designed for VAG (Volkswagen Audi Group). Therefore, it provides numerous sensor data in the car. In VCDS, Advance Measuring Values function gives selected up to 12 measuring values at the same time.

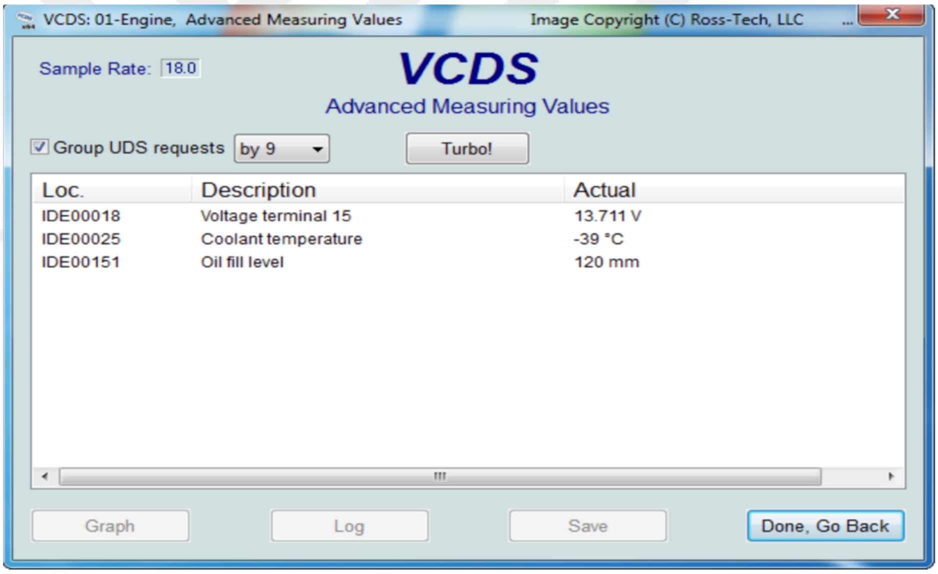

Figure 3.3 VCDS Advanced Measuring Screen[10]

3.6.2. Vehicle Data

Collectable data information from vehicles is explained as follows.

---

[9] http://www.ross-tech.com/index.html

[10] http://www.ross-tech.com/vcds/tour/adv-meas-blocks.php

Engine Speed

It is also known as revolutions per minute (RPM) (Hunting, 2017). The speed is at which the engine revolves internally. In other words, RPM is a number of crankshaft's full rotation each minute. The crankshaft is connected to pistons by rods. Depending on the position of the crankshaft, the pistons are moving up and down in its cylinder.

Engine Load

Although engine load is a standard UDS parameter among vehicle producers, there is not a standard definition of it.

One of the definitions of engine load (Kean, et al., 2003) is the amount of external mechanical resistance against turning a crankshaft. There are several engine load sources, which are wind resistance, tire-roadway friction, vehicle acceleration, roadway curve, engine friction, and use of accessories such as air conditioning.

Another definition from SAE J1979 (Alessandrini, et al., 2009) is the amount of current torque through the engine as a percentage of the theoretical maximum torque.

$$Engine\_Load = \frac{Current\_Torque}{Max\_Torque(RPM).\frac{Baro}{29.92}\cdot\sqrt{\frac{298}{T(amb)+273}}}$$

Equation 3.1 Engine Load

Vehicle Speed

Vehicle speed is what driver sees on the dash panel as wheel rotation speed.

Intake Air Temperature

The Intake Air Temperature is used to calculate fuel efficiency in the engine control unit. It is the temperature of the air that is entering the engine of a vehicle. The oxygen in the air is required to burn fuel (Setiawan, 2018).

Boost Pressure

Boost pressure indicates manifold air pressure in an internal combustion engine [11].

Ambient Air

According to the fuel economy tests, vehicles consume in short distance trip about %12 more fuel at cold days [12]. The vehicle consumes extra fuel because of several reasons. The most important of them is engine and transmission friction increases in cold temperatures and the vehicle takes a longer time to reach the most fuel-efficient-temperature.

Engine Torque

Engine Torque is the value of rotational force applied on the crankshaft by the piston in an engine [13]. Its unit is Newton-meter (Nm).

The engine torque is very important for acceleration and performance at load. It is related to horse-power by speed. They are both outputs of an engine. Torque is the demanded pulling ability of the vehicle.

---

[11] https://en.wikipedia.org/wiki/Boost_gauge
[12] https://www.fueleconomy.gov/feg/coldweather.shtml
[13] https://carbiketech.com/engine-torque/

Torque Specified

Torque specific is the expected maximum torque specified by the ECU.

Brake Electronics

VCDS provides information about several brake electronics those are ABS, EDS, ASR and ASC. Definitions of them are given below:

Anti-lock braking system (ABS) is a safety anti-skid braking system on vehicles. The system prevents locking of the wheels during braking suddenly. Thus, it enables less braking distance for vehicle [14].

Electronic Differential System (EDS) is an electronic differential lock which adjusts torque amount for each driving wheel and provides different speeds to each wheel while cornering safely [15].

Anti-Slip Regulation (ASR) is also known as Transaction Control System (TCS). The ASR helps prevent the vehicle's wheels from losing traction by using an electro-hydraulic system as a secondary safety function. It gets involved when throttle input and engine torque are in conflict to road frictions [16].

Electronic stability control (ESC) is also known as *electronic stability program (ESP)* or *dynamic stability control (DSC)*. The computerized system is designed to help a vehicle's stability by detecting and reducing the loss of traction. When any loss of steering control is detected, it automatically applies the brakes to the wheel being need of help. Some ESC systems also decrease engine power until the driver regains control [17].

[14] https://www.scienceabc.com/innovation/abs-sensors-anti-lock-breaking-system-technology-cars-work.html
[15] https://en.wikipedia.org/wiki/Electronic_differential
[16] https://auto.howstuffworks.com/28000-traction-control-explained.htm
[17] https://www.wisbusiness.com/2008/farmers-insurance-introduces-auto-insurance-electronic-stability-control-esc-discount-in-wisconsin/

Steering Angle

The steering angle is an angle between the front of the vehicle and the steered wheel direction. The parameter value is generally used for diagnostic operations.

Coolant Temperature

A coolant keeps the engine at optimum temperature. This temperature data is used by an engine control unit to adjust the fuel injection and ignition timing. In other words, an engine requires more fuel when it is cold and less fuel when the temperature is high.

Oil Temperature

The oil temperature is a critical value for your engine health. The oil temperature should be a few degrees warmer than the coolant. It is also important for fuel consumption. Reducing warming up time of oil provides less fuel consumption and emission (Cipollone, et al., 2015).

Injecting Timing

Injecting timing is the time value of when fuel is discharged into the cylinder. This timing is usually balanced to get as much power as possible, while still remaining in legal limits for emissions [18].

Mass Air Flow

The Engine Control Unit (ECU) uses mass air flow rate value to balance and deliver the correct fuel mass to the engine. The engine's air-fuel ratio is adjusted more accurately when a MAF sensor is used in addition to the oxygen sensor [19].

---

[18] https://highwayandheavyparts.com/n-12787-adjusting-diesel-engine-injection-timing.html
[19] https://en.wikipedia.org/wiki/Mass_flow_sensor

Throttle Valve Angle

The throttle valve is located in the intake air system of the combustion engine. The opening angle of the valve arranges how much fresh air or air/fuel mixture flows into the cylinders. For fuel efficiency, the injected fuel and air must be matched near-ideal [20].

Lambda Value and Correction Factors

Air/Fuel (A/F) ratio is controlled by the engine control unit (ECU) to maintain power, efficiency and emissions. A/F is expressed as a ratio or as a Lambda value [21]. Lambda 1.0 value is equal to 14.7:1 A/F ratio with iso-octane (ideal gasoline).

ECU has described values for a given RPM, A/F ratio, load etc. Controlling of the A/F ratio is provided with controlling of injectors. ECU snitches, which are oxygen sensors and the MAF, reports desired A/F ratio has been attained or not. If the ratio is not attained as expected, ECU applies a correction factor to the fuel injectors for attaining to desired A/F ratio. The correction ratio and lambda value can be readable with VCDS software.

Accelerator Pedal Position

Position information of the accelerator (throttle) pedal is transmitted continuously to the engine control unit (ECU) by accelerator pedal sensors [22]. Requested load by a driver can be applied immediately based on information of these sensors. It is possible to access these sensors data with the VCDS software.

---

20 https://www.bosch-mobility-solutions.com/en/products-and-services/commercial-vehicles/powertrain-systems/natural-gas/electronic-throttle-valve/

21 https://www.seat.com/car-terms/l/lambda-control.html

22 https://www.hella.com/techworld/uk/Technical/Sensors-and-actuators/Accelerator-pedal-sensor-3851/

Kick-Down

Kick-Down[23] is a special behaviour of the throttle pedal. Especially automatic transmission vehicles have this ability. Kick-Down is triggered when the driver pushes the throttle pedal to the floor. When Kick-Down is triggered, transmission shifts down if possible. Hence vehicle can accelerate as quickly as possible. Kick-Down is supported by Most Drive-by-Wire (DBW) throttle systems. Count of triggered Kick-Downs is accessible with the VCDS software.

Generator Load

Generator load, is the consumed electrical power by the connected elements to the generator, can be accessible with the VCDS software.

Idle Speed and Idle Speed Control Valve Position

Idle speed is the engine rotational speed value when the throttle pedal is not pressed. Revolutions per minute (rpm) is the unit of the idle speed. Regulation of the idle speed is provided by the Idle Speed Control (ISC) valve. That valve is also known as the Idle Air Control (IAC) valve. Both multipoint fuels injected, and throttle body engines have the idle speed control valve. This valve is called an Automatic Idle Speed (AIS) motor for Chrysler brand vehicles. It is known as Idle Speed Control (ISC) solenoid for the Ford brand vehicles.

[23] https://www.cars.com/articles/what-does-kickdown-mean-in-an-automatic-transmission-1420690417731/

4. SOLUTION METHODOLOGY

In this chapter, the goal is to create a machine-learning model that will be used for observing the effect of any modifications on vehicle to reduce fuel consumption easily. First of all, vehicle sensor data is analysed and validated that its calculated fuel consumption value is consistent. Then, the proposed solution for predicting fuel consumption will be discussed. Finally, tests and results of the algorithm will be presented.

4.1. Manual Data Analyses

After the first data collection attempt is completed, the data was examined. Although it does not have a sufficient number of instances to build a successful model, it can be used for experimental work. It has 642 rows and each row has instantaneous engine speed, engine load, injection timing, throttle valve angle, ignition timing angle, battery voltage, intake air temperature, fuel consumption signal, fuel consumption equivalent, vehicle speed, boost pressure actual and engine torque actual features. MS Excel was used to perform the initial visualization of the data. Simplified representation of data is shown at Table 4.1.

Table 4.1 Collected Data Example

Engine Speed	Engine Load	Injection Timing	Throttle Valve Angle	Ignition Timing Angle	IntakeAir Temp	Fuel Cons. Eq	Vehicle Speed	Boost Pressure Actual	Engine Torque Actual
840	25.6	1.28	3.9	-4.5	45	12	0	980	13.6
840	24.1	1.28	4.3	-5.3	45	12	1	980	9.7
840	21.8	1.28	3.9	-4.5	45	22	2	980	3.9
840	21.8	1.02	3.9	-3.8	45	22	3	980	5.8
840	19.5	1.02	3.5	-5.3	45	20	3	980	1.9
840	18.8	1.02	3.5	-5.3	45	20	4	980	1.9
760	18.8	1.02	3.1	-6	45	16	3	980	1.9
720	21.1	1.02	3.1	-0.8	45	16	0	980	1.9

1200	28.6	1.53	3.1	-1.5	45	16	0	980	21.3
1200	37.6	1.53	7.5	14.3	45	16	0	980	50.4
1120	36.1	1.79	7.8	14.3	44	16	4	990	48.5
1480	28.6	1.28	7.5	12.8	42	40	8	990	46.6
1640	26.3	1.53	6.7	21	41	40	10	990	21.3
1160	37.6	1.53	8.6	-0.8	41	46	11	1000	52.4
1280	30.1	1.28	7.8	12	41	46	13	1000	48.5
1280	25.6	1.28	5.9	17.3	40	46	14	990	25.2
1360	23.3	1.28	6.3	18.8	40	46	15	990	25.2
1360	17.3	1.02	5.1	21	40	46	16	990	5.8

Figure 4.1 shows that the vehicle is initially at an idle state for a while (engine speed ~ 760 - 840 rpm) and engine torque and the vehicle speed is 0 as expected. Then, the vehicle moves with some torque and the fuel consumption equivalent also changes with this torque.

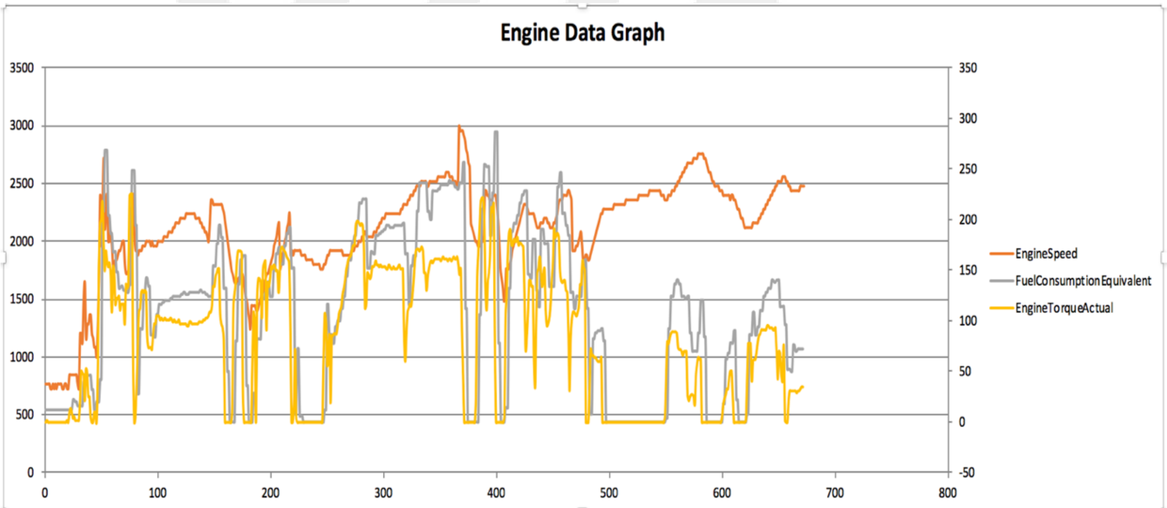

Figure 4.1 Engine Data Graph

Before applying machine learning algorithms to the dataset, some useful calculations are done on the dataset. First of all, distance is calculated as a new feature by multiplying the difference between two time-stamps with vehicle speed. Then, it is divided by 3600 to obtain km. Instant fuel consumption is evaluated by multiplying time stamp difference with adjusted fuel consumption equivalent. Finally, value is divided by 3600 to obtain L/h. Instantaneous fuel consumption L in 1 km (L/km) is evaluated by dividing fuel consumption (L) to distance (km).

Table 4.2 Calculations for Total Values from Data Set

Total Distance	Total Consumption	Average L per km	Average L per 100 km
16.03626 km	2.018137	0.125848 L / km	12.5848 L / 100 km

The vehicle travelled 16.03 km and consumed 2.018137 L in total. It shows that the vehicle consumes average 0.12584 L for this travel, and we can assume that the vehicle will consume 12.584 L in 100 km. We realized that the test vehicle consumes one-tenth of our total consumption calculation in real world. Therefore, we can consider that fuel consumption equivalent is ten times of actual fuel consumption. This analyse shows that we can use fuel consumption equivalent as our target value. Hence, we concluded that the value is coded with a fixed decimal point.

Manually estimating the results even when consulting experts in the automotive field was a nontrivial solution. Hence, some machine learning techniques were employed to overcome the problem seamlessly.

4.2. Predicting Fuel Consumption

To achieve a solution, machine learning methods were used to predict fuel consumption by processing and training vehicle sensors' data. Our main aim is to predict fuel consumption value according to the CanBus data. Therefore, a regression model is built for this purpose.

4.2.1. Machine Learning

Arthur Samuel described machine learning as: “the field of study that gives computers the ability to learn without being explicitly programmed.” (Puget, 2016).

(Mitchell, 1997) provides a more modern definition as: “A computer is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Let's give an example of a chess game:
E = the experience of playing many checkers
T = the task of playing checkers
P = the probability to win the next game

Also, machine learning is defined as "Programming computers to optimize a performance criterion using example data or past experience." by (Alpaydin, 2016).
This is important to note that machine learning highly depends on the theory of statistics in providing mathematical models to come up with inference from a sample. Types of Machine Learning are divided into supervised and unsupervised learning. In the case of knowing the relationship between the input and output within a given dataset, supervised learning can be used to predict the future output (regression). We used the supervised learning because of the fact that we tried to predict fuel consumption using all vehicle CanBus data as input.

Regression Example

A given data set includes the size of the houses and their prices. The price will be predicted as a function of size as a continuous output as shown in Figure 4.2.

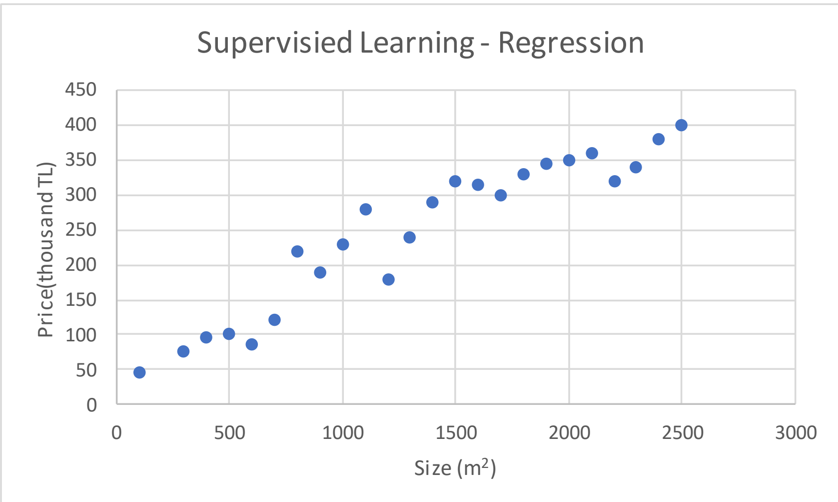

Figure 4.2 Supervised Learning Regression Example

As the example above, in our case we are predicting fuel consumption equivalent as our target value.

4.2.2. Machine Learning Tools Utilized

Gradient boosted trees, XGBoost and Artificial Neural Network are three known machine learning algorithms. They are chosen because of their execution speed and also, they do not need so much data for training except of ANN. Besides, ANN is experimented because it does not need fully completed data and its performance is observed with using existent dataset. In this thesis, they are benefited from analysing CanBus data.

4.2.2.1. Gradient Boosted Trees Algorithm

The Gradient Boosted trees algorithm is a group of learning methods that combines a lot of decision tree algorithms for calculation the prediction. It is used for classification and also regression. It produces a prediction model in the form of an ensemble of weak prediction models. These models are typically decision trees. It generalizes models by optimization of a function of arbitrary differentiable loss [24].

The “tree” part of gradient boosted trees refers to the fact that the final model is a forest of decision trees. “boosted” part means that a group of week learners can be merged to produce a strong learner. The “gradient” part of the algorithm refers to the method by which the new function is added to the model (Korolev & Ruegg, 2015).

Gradient boosted trees algorithms can be used for ranking. Yahoo and Yandex, which are the commercial web search engines, use some variants of gradient boosted trees algorithms in their machine-learned ranking engines.

4.2.2.2. XGBoost

XGBoost (eXtreme Gradient Boosting) is an open-source algorithm which implements gradient boosted decision trees [25]. It is an optimized distributed gradient boosting library

---

[24] https://github.com/yarny/gbdt
[25] https://github.com/dmlc/xgboost

which is designed to be highly efficient, portable and flexible [26]. It works on Linux, Windows and macOS and can be used with C++, Java, R, Julia programming languages. It also has support for the distributed processing frameworks like Hadoop, Spark, and Flink. It provides a parallel tree boosting (known as GBRT - Gradient Boosted Regression Tree, GBM – Gradient Boosted Machines) that solve many data problems.

XGBoost is a popular library in supervised machine learning. The most important point of the success of XGBoost is its scalability and performance. It runs much faster than existing popular solutions on a single computer. The scalability of it is due to several important algorithmic and systematic optimizations. Parallel and distributed computing make data learning much faster and calculating model much quicker (Chen & Guestrin, 2016).

4.2.2.3. Neural Network and Deep Learning

A neural network is the connection of simple processing elements, units or nodes, whose functionality is loosely based on animal neurons. Neural network processing ability comes from interunit connection strengths and weights, the process of adaptation, training (Gurney, 2004).

Neural networks can be used in lots of area such as forecasting time series, signal classification and pattern recognition. A neural network is a model which is inspired by human brain. It consists of multiple connected neurons [27]. The network consists of three layers [28]. These layers are input layer that takes existing data, a hidden layer that uses backpropagation to optimize the weights of the inputs in order to improve the prediction capability of the model, and an output layer that evaluates prediction data from the input and hidden layer data.

---

[26] https://xgboost.readthedocs.io/en/latest/

[27] https://blog.webkid.io/neural-networks-in-javascript/

[28] https://datascienceplus.com/keras-regression-based-neural-networks/

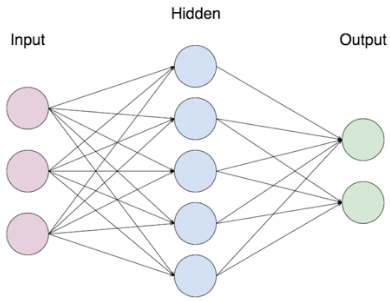

Figure 4.3 Simple Neural Network Layers

Deep Learning is a field of machine learning based on algorithms which have the structure and function of neural networks (Brownlee, 2016). Deep learning represents the most recent point of artificial intelligence (AI). With deep learning, the process is not taught to a computer. Instead of it, computer trains itself about the process and with given data. A deep learning system is a self-teaching system which filters information through multiple hidden layers like humans.

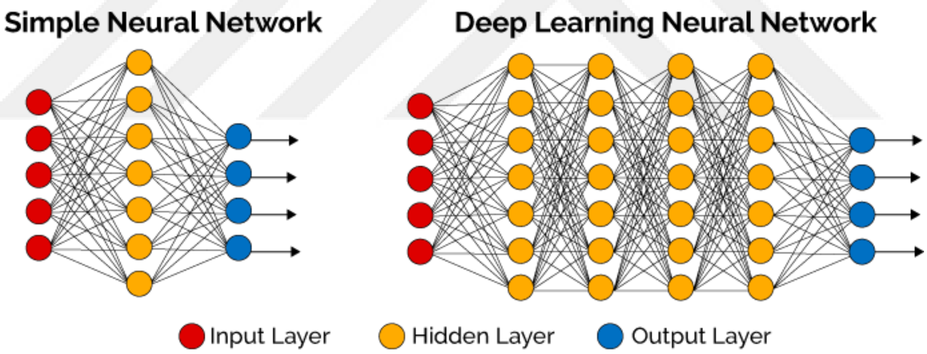

Figure 4.4 Simple Neural Network and Deep Learning Neural Network Layers [29]

Deep learning is the technology behind driverless vehicles. It is the technology for controlling consumer devices like phones, tablets, smart TVs, personal computers and hands-free speakers with only voice of user [30]. Deep learning is getting lots of attention nowadays in much more part of technology.

---

[29] https://www.linkedin.com/pulse/intuition-deep-neural-networks-abhishek-parbhakar/

[30] https://www.mathworks.com/discovery/deep-learning.html

4.2.2.4. Knime Analytics Platform

KNIME [31] Analytics Platform is one of the most popular, open source software tools for creating data science services and application. Knime integrates various components for data mining and machine learning through its modular data pipelining concept. A graphical user interface with an intuitive, drag and drop style allows assembly of nodes creating visual workflows with or without only minimal programming. Knime is implemented in Java programming language but also allows for wrappers calling another coding language in addition to providing nodes that allow running Java, Python, Perl and other code fragments.

4.2.3. Model Training and Data Set

In this phase of the study, the procedure was started by logging training and testing vehicle data. They were logged at different time, weather and road conditions using our test vehicle Passat 1.8 TSI. Then, all log files were concatenated in one file.

Knime Analytics Platform was used to visually implement algorithms and familiarize with the dataset. Our main aim for pre-processing was making our data ready for training. An information gain table is obtained using the Knime Information Gain Calculator node.

The information gain is a metric used to train Decision Trees. The metric measures the quality of a split. Constructing a decision tree is all about finding attribute that returns the highest information gain.

[31] https://www.knime.org/

Table 4.3 Information Gain Table

S featureName	D informationGain
Boost Pressure – (actual)	1.279
Throttle Valve Angle	1.173
Engine Load	1.062
Engine Torque – (actual)	0.971
Injection Timing	0.906
Engine Speed – (G28)	0.685
Vehicle Speed	0.497
Intake Air – Temperatu...	0.286

According to the Table 4.3, the most important feature for forecasting fuel consumption is Boost Pressure. Throttle Valve Angle, Engine Load, Engine Torque, Injection Timing, Engine Speed, Vehicle Speed and Intake Air – Temperature follows Boost Pressure respectively.

We built a Multilayer Perceptron Neural Network model using Knime Analytics tool as follows. First of all, the unified csv log file was loaded to application and pre-processing was completed. At the pre-processing phase, header and unit rows were removed from the dataset. Then, unused timestamp fields and irrelevant features are removed. All remaining fields were normalized as 0 to 1. After the normalizing process, we divide into a dataset for training and testing. Since we had 15000 rows data approximately, we split into 3000 rows for testing and remaining 12000 rows for training. After experiencing various combination of iterations, hidden layers and neurons, the best performance in terms of accuracy has been obtained for our model under a number of iterations as 100 and designed 4 hidden layers and 10 neurons for each layer.

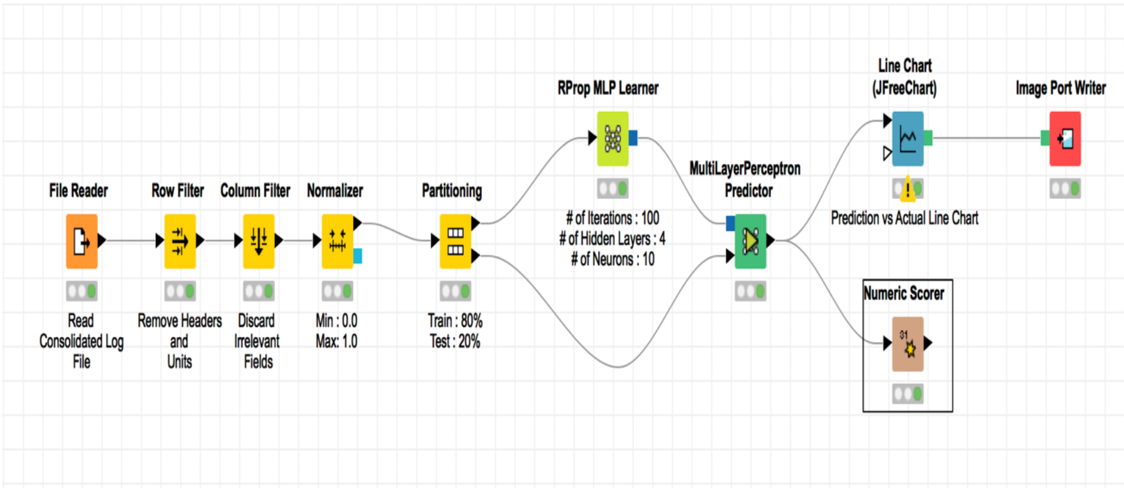

Figure 4.5 Multilayer Perceptron Neural Network Process on Knime

To see the results of the related algorithm performance, a numeric scorer and line chart were added to the output of the predictor. Figure 4.6 shows only first 500 rows result to see the differences and similarities between target and predicted fuel consumption value.

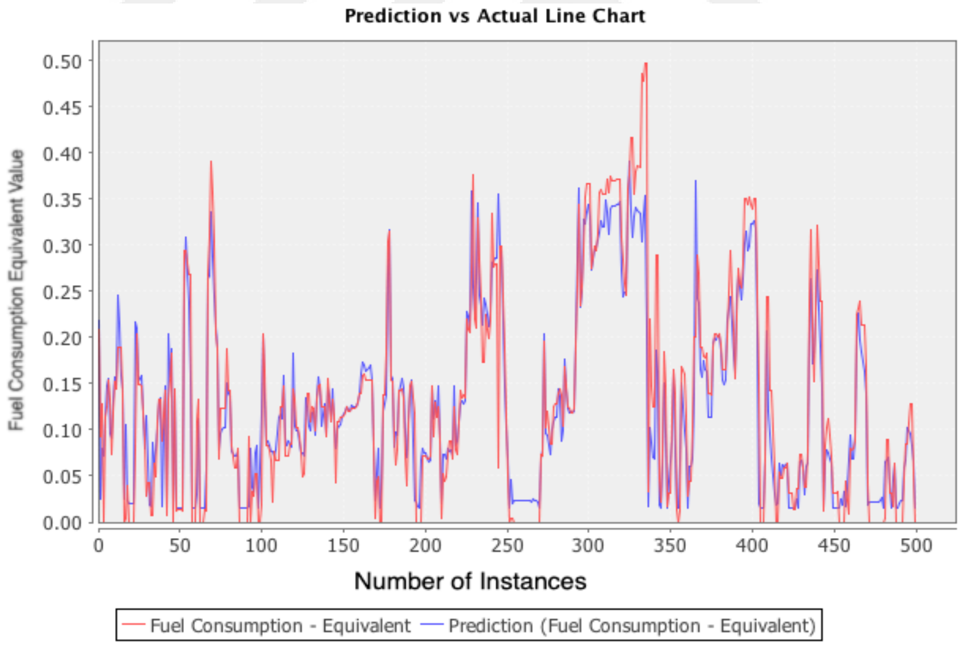

Figure 4.6 Target vs Predicted Fuel Consumption Line Chart for Multilayer Perceptron Neural Network

Multilayer Perceptron Neural Network algorithm gave R result as 0.826 and 0.022 mean absolute error. Mean absolute error seems good but it should not be forgotten that predicted value was normalized 0 to 1 range.

R^2:	0.826
Mean absolute error:	0.022
Mean squared error:	0.001
Root mean squared error:	0.039
Mean signed difference:	0.001

Figure 4.7 Test Results for Multilayer Perceptron Neural Network

As a second algorithm, we established a Gradient Boosted Trees algorithm using Knime Analytics tool as follows. First of all, we applied similar pre-processing, normalization and partitioning process like the previous algorithm. The best result is obtained for our fuel consumption model with a number of tree depth as 4, 0.05 learning rate and 250 models as boosting options.

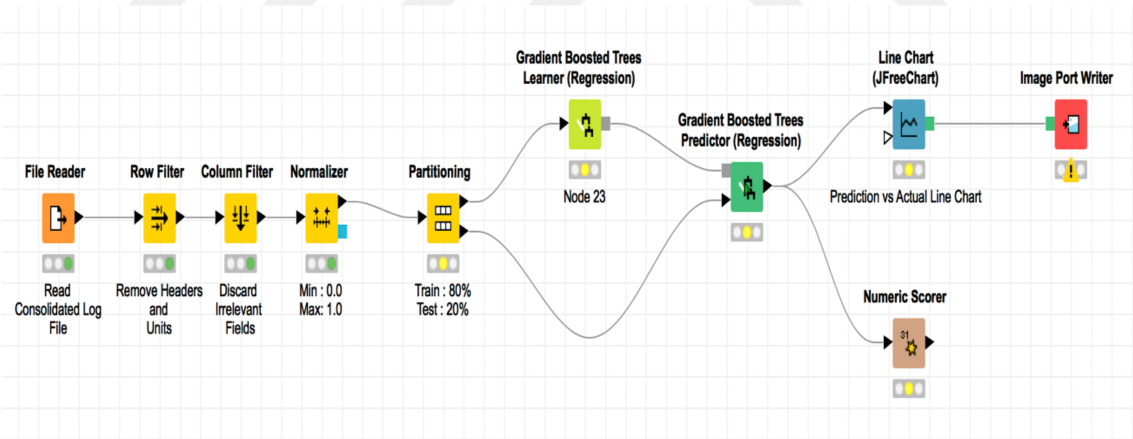

Figure 4.8 Gradient Boosted Trees Algorithm Process on Knime

Figure 4.9 shows only first 500 rows result to see the differences and similarities between target and predicted fuel consumption value.

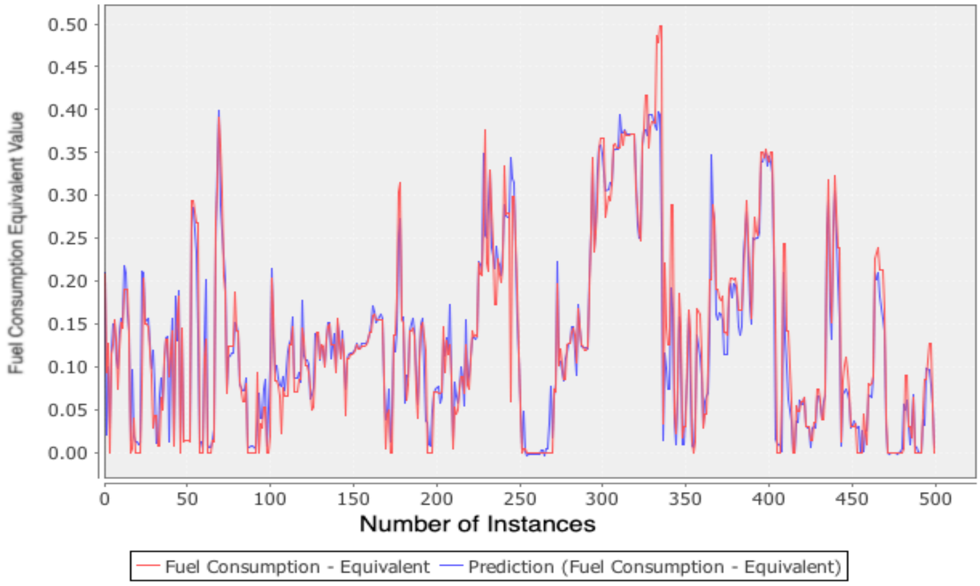

Figure 4.9 Target vs Predicted Fuel Consumption Line Chart for Gradient Boosted Trees Algorithm

Gradient Boosted Trees Algorithm gave R result as 0.888 and 0.017 mean absolute error. Gradient Boosted Trees Algorithm seems working better than ANN.

R^2:	0.888
Mean absolute error:	0.017
Mean squared error:	0.001
Root mean squared error:	0.032
Mean signed difference:	0

Figure 4.10 Test Results for Gradient Boosted Trees

The last algorithm we designed is XGBoost for forecasting vehicle fuel consumption. Similar pre-processing, normalization and partitioning process were applied such in the previous algorithm to be ready for model training and testing. In Knime tool, XGBoost tree was used with its default values as a booster, 6 as maximum depth, 1 as maximum child depth, 0.3 as eta value and 100 as boosting rounds generally for our data set and model, the algorithm gave the best results with configuration above.

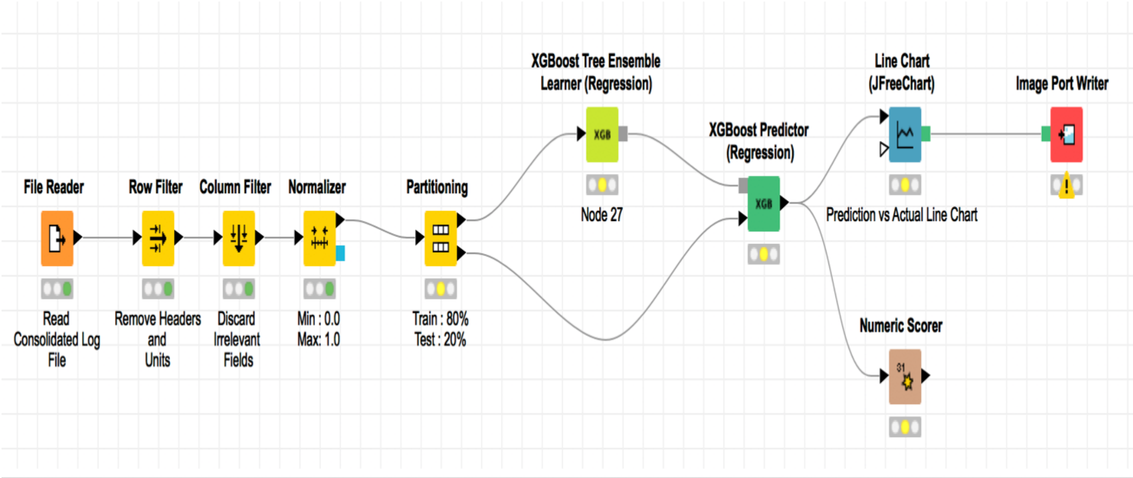

Figure 4.11 XGBoost Process on Knime

Figure 4.12 shows only first 500 rows result to see the differences and similarities between target and predicted fuel consumption value.

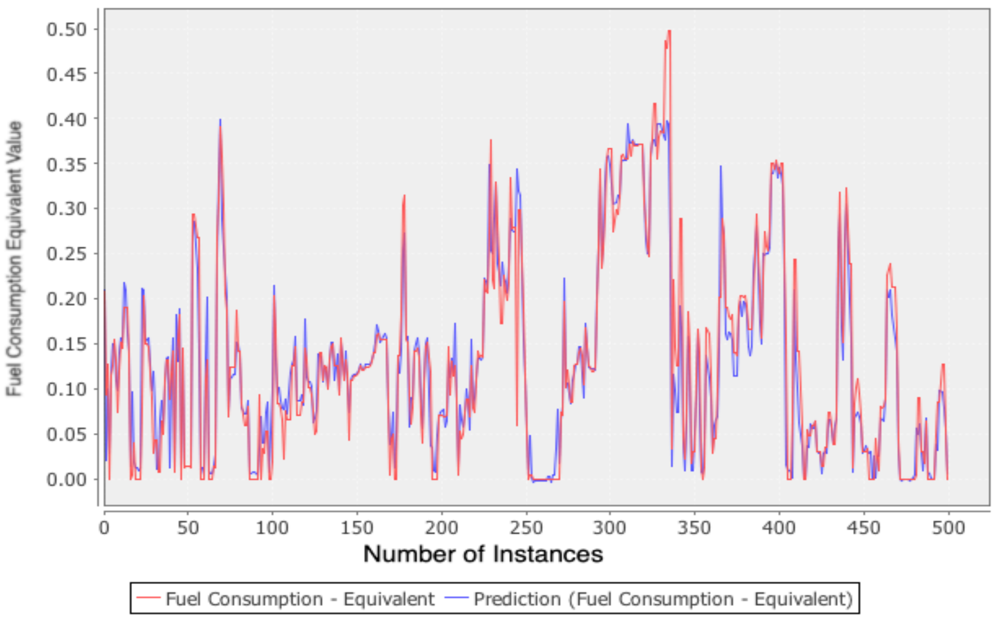

Figure 4.12 Target vs Predicted Fuel Consumption Line Chart for XGBoost

XGBoost Algorithm gave R result as 0.891 and 0.017 mean absolute error.

R^2:	0.891
Mean absolute error:	0.017
Mean squared error:	0.001
Root mean squared error:	0.031
Mean signed difference:	0.001

Figure 4.13 Test Results for XGBoost Algorithm

XGBoost Algorithm is also tested without normalizing all features. Normalization is not a necessity for XGBoost algorithm, because of the fact that numeric score gave better results in that way.

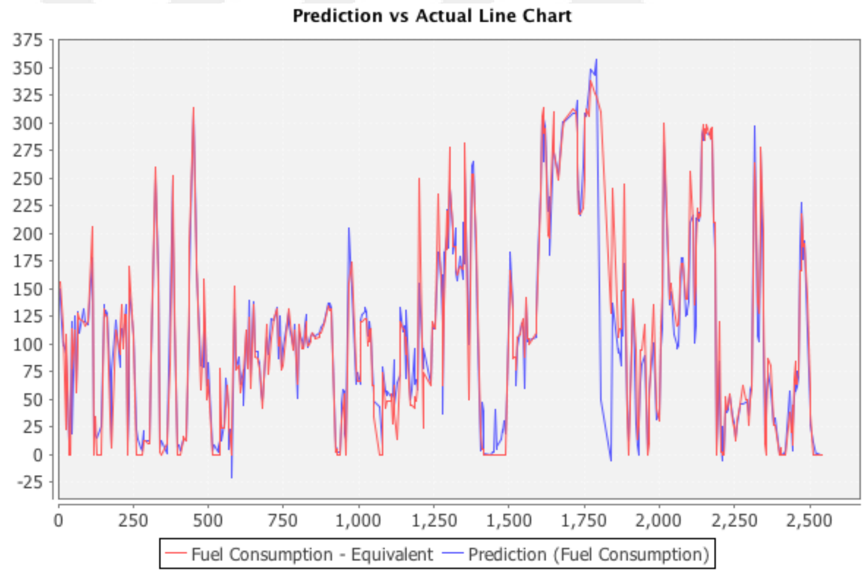

Figure 4.14 Target vs Predicted Fuel Consumption Line Chart for XGBoost (without normalization)

XGBoost Algorithm without normalizing gave R^2 result as 0.902 and 13.262 mean absolute error. According to the numeric scorer results, XGBoost Algorithm (without normalization features) is the best suitable solution for our dataset.

R^2:	0.902
Mean absolute error:	13.262
Mean squared error:	602.91
Root mean squared error:	24.554
Mean signed difference:	0.021

Figure 4.15 Test Results for XGBoost Algorithm (without normalization)

4.2.4. Tests and Results

When all target vs predicted fuel consumption line charts are examined, it can be seen that in some cases there is too much difference between the estimated value and the actual value. The reason for these differences; when there are sudden changes in fuel consumption, it is the inability to accurately sample the values that cause sudden changes. Because the vehicle data transfer sampling rate is not good enough to sense these sudden changes.

The coefficient of determination, or R^2 ranges between “0” and ”1”, with “0” indicating that the built model does not improve prediction over the mean model, and “1” indicating perfect prediction. While the model is improving, R^2 results increases proportionally. When R^2 is negative it means that the model is worse than predicting the mean. Therefore, R^2 is the most suitable metric for our problem because it is scale-free, and the algorithms tried for our model are used both scaled and non-scaled data-set to observe performances of them.

$$R^2 = 1 - \frac{\mathrm{MSE(model)}}{\mathrm{MSE(baseline)}}$$

Equation 4.1 The Coefficient of Determination (R^2)

Mean Absolute Error measures the average magnitude of the errors in a set of predictions, without considering their direction positive or negative. It’s the average error between the expectation and actual value.

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N} |y_i - \hat{y}_i|$$

Equation 4.2 Mean Absolute Error

The Root Mean Squared Error is the square root of the variance of the residuals. It indicates the absolute fit of the model to the data–how close the observed data points are to the model's predicted values. Lower values of RMSE indicate better fit. RMSE is a good measure of how accurately the model predicts the response.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y})^2}$$

Equation 4.3 Root Mean Square Error

Table 4.4 Comparison for Algorithms

Algorithm	**R^2**	**MAE**	**RMSE**
Multilayer Perceptron Neural Network	0.826	0.022	0.039
Gradient Boosted Trees	0.888	0.017	0.032
XGBoost	0.891	0.017	0.031
XGBoost (Without Normalized Features)	0.902	13.262	24.554

MAE and RMSE results for our all algorithms except for XGBoost (Without Normalized Features) gives the best result for XGBoost. Since we used normalized and non-normalized data-set for our algorithms, considering R^2 result is more meaningful. According to the results, XGBoost without normalized features produces the best R^2. Therefore, we use the XGBoost algorithm with the configuration mentioned above without normalization.

5. FIELD TESTING AND OBSERVATIONS

Once a successful model is obtained according to the performance results, we need to obtain modified vehicle CanBus data to observe the effect of modification on fuel consumption. Several test cases are organized to demonstrate gains or losses in the system after modification using the software ability. Following test scenarios are selected because their effects on the car are almost predictable and testing them easy in short time. Therefore, these test scenarios are examined at field testing phase of this study.

5.1. Test Scenarios

The test cases include changing aerodynamic, drag ability, vehicle speed and careless driving habit. Many drivers have a suspicion that opened windows while driving effects fuel consumption negatively. Therefore, we drove with fully opened and closed windows at different vehicle speeds to observe the effect of an aerodynamic change on the car. Secondly, some drivers use their car carelessly. Although the car has higher gear options, some drivers don't use the convenient gear selection because of carelessness. To see the effect of this case, we drove at the same route with only sixth (the test car has seven gear options) gear at desired high speed. As a third case, we drove with a low and high pressure tire at different speeds. People believe the lower tire pressure the higher fuel consumption. Our aim is to reveal this assumption is right or not with this test case. In addition to these, people consider that fuel consumption increases after a certain speed. Therefore, drivers who want to spend less money on fuel, tend to drive below that certain speed to decrease fuel consumption.

5.2. Test Cases

A file name format is determined for identifying test case log file as *tire-pressure_vehicle-speed_window-state_nth-gear*. For example, tire pressure is 22, the vehicle speed is 80, windows are closed and gear with automatic transmission (Drive mode). For this test case, the file name should be *22_80_Closed*. In this way, we had eight test cases as follows: *22_80_Closed, 22_120_Closed, 50_80_Open, 50_80_Closed, 50_120_Open, 50_120_Closed_Gear6, 50_120_Closed* and *50_160_Closed.*

Table 5.1 Test Cases - File Name Table

ID of Test Case	***Tire Pressure (PS)***	***Vehicle Speed (km/h)***	***Window State***	***# of Gear***	***File Name***
Case-1	22	80	Close	Auto	22_80_Closed_LOG.csv
Case-2	22	120	Close	Auto	22_120_Closed_LOG.csv
Case-3	50	80	Open	Auto	50_80_Open_LOG.csv
Case-4	50	80	Close	Auto	50_80_Closed_LOG.csv
Case-5	50	120	Open	Auto	50_120_Open_LOG.csv
Case-6	50	120	Close	6	50_120_Closed_Gear6_LOG.csv
Case-7	50	120	Close	Auto	50_120_Closed_LOG.csv
Case-8	50	160	Close	Auto	50_160_Closed_LOG.csv

5.3. Data Gathering and Pre-processing

We gathered data after midnight at Istanbul/Kocaeli highway and drove at the same route for different conditions which are mentioned in the previous section since environmental factors should not affect our observations. The trip distance was approximately 200 km at the same route and took more than 4 hours. Same test vehicle VW Passat 1.8 TSI and VCDS data logging tool were used to collect data for several cases.

Vehicle's cruise control system was used to stabilize the speed at the desired level. Reaching out to desired speed with the cruise control system is a time-consuming process.

Therefore, lower speed values were logged until that time. At the pre-processing phase, these irrelevant records with lower vehicle speed were removed from the dataset.

Moreover, we realized that the vehicle reached out maximum 75 km/h exactly however speedometer shows 80. Therefore, we decided to use a threshold for vehicle speed value.

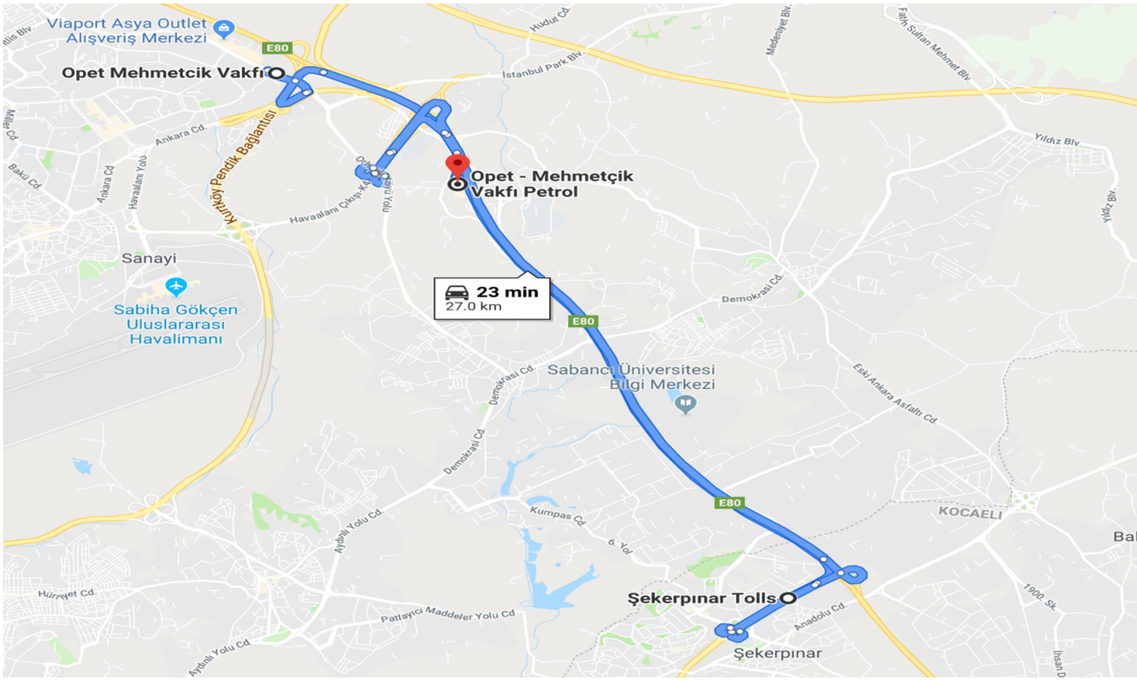

Figure 5.1 Test Cases Trip Route

For example, we assume that if the test is for 80 km/h, we removed records with vehicle speed lower than 70.

5.4. Analysis of Tests Cases

Normally, the training data for obtaining a model should be collected with a car’s stock condition for fuel consumption to compare the modifications impact on the car healthfully. However, the model was trained with the data which are collected at different times and conditions (route, tire pressure, windows state, vehicle speed, air condition state, etc.) because of lack of time. Therefore, these values are not the stock values of the car for fuel consumption. After all, the trained model with these data gave successful performance results as we also mentioned in the previous section in details.

A stock condition is determined as an assumption by selecting the value at the middle of the sorted actual fuel consumptions list of the test cases. In our study, the car consumes average 120 units as the fuel consumption equivalent (real-like 12 L) with 50 PS, 120 km/h and closed windows (Case-7) which is assumed its stock condition.

Table 5.2 Sorted Actual Fuel Consumption List

Test Cases	***Actual Fuel Consumption***
50_80_Closed	55.462
50_80_Open	56.375
22_80_Closed	64.579
50_120_Closed (Case-7)	120.629
22_120_Closed	128.476
50_120_Open	131.473
50_120_Closed_Gear6	135.253
50_160_Closed	260.847

The details of each test cases which are given Table 5.1 are discussed under the appropriate subtitles below by their case id. Since the fuel consumption equivalent is not a real consumption of vehicle (L), we couldn't use any unit on following tables and graphs.

5.4.1. Case-1

In Case-1, we deflated the tires 22 PS, and set cruise control speed to 80 km/h and closed all windows. We drove our test car with Drive mode in DSG (automatic transmission of VW). We logged our all trip data in a csv file as 22_80_Closed_LOG.csv.

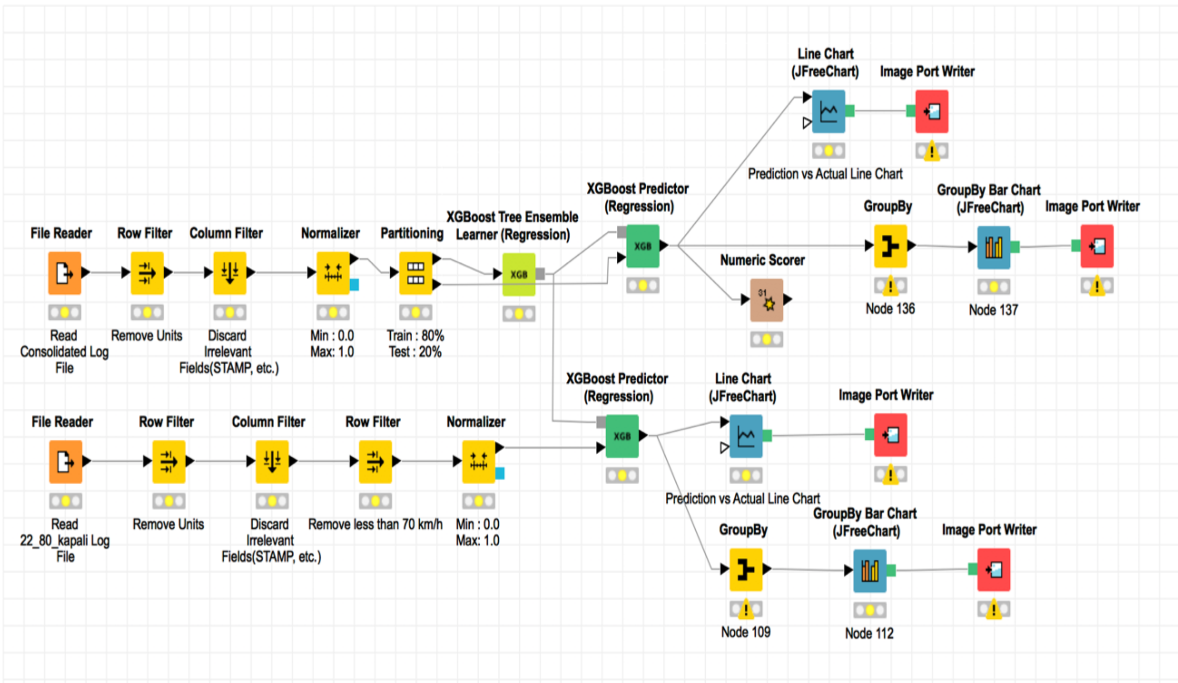

Figure 5.2 Model Representation of Case-1 on Knime

While the car dashboard showed the speed as 80 km/h, the CanBus gave the speed as 76 km/h. In addition to these, we decided to filter out records with speed less than 70 km/h to provide a realistic approach.

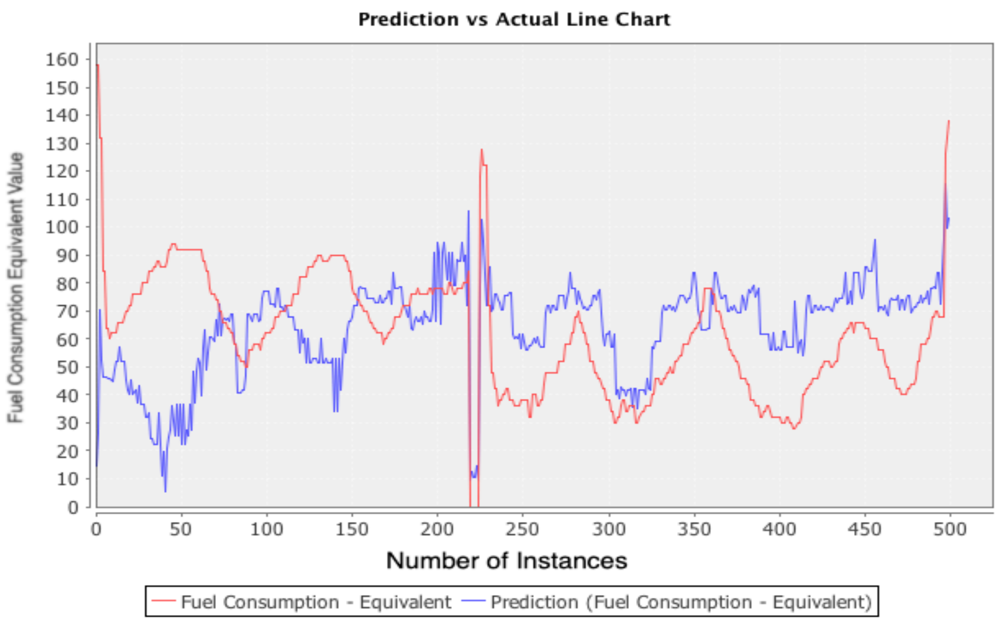

Figure 5.3 Case-1 - Values for Prediction vs Actual Fuel Consumption

5.4.2. Case-2

In Case-2, we continued with the tire pressure 22 PS, and set cruise control speed to 120 km/h and closed all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 22_120_Closed_LOG.csv.

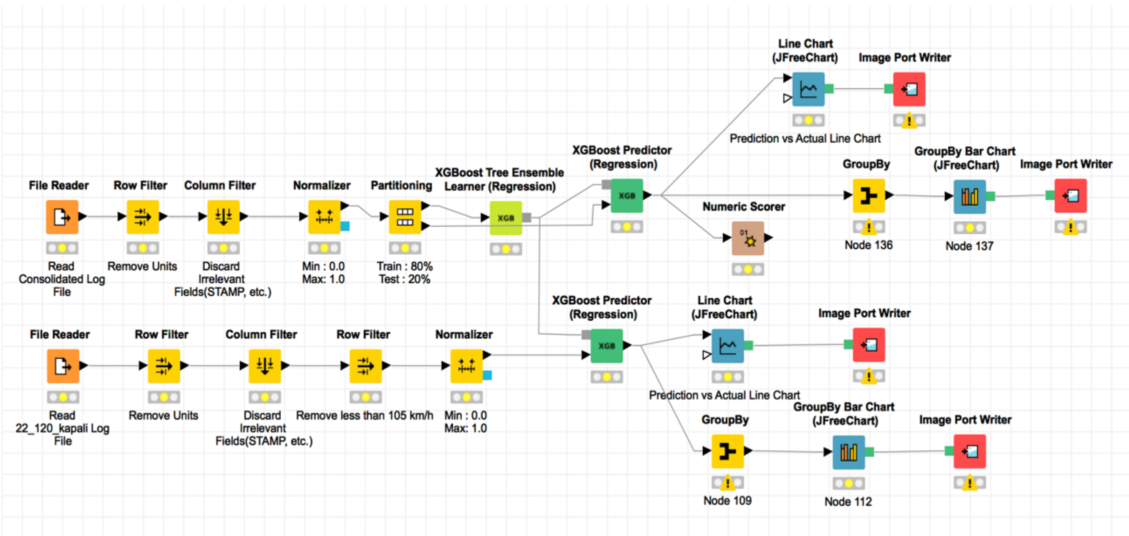

Figure 5.4 Model Representation of Case-2 on Knime

While the car dashboard showed the speed as 120 km/h, the CanBus gave the speed as 114 km/h. In addition to these, we decided to filter out records with speed less than 105 km/h to provide a realistic approach.

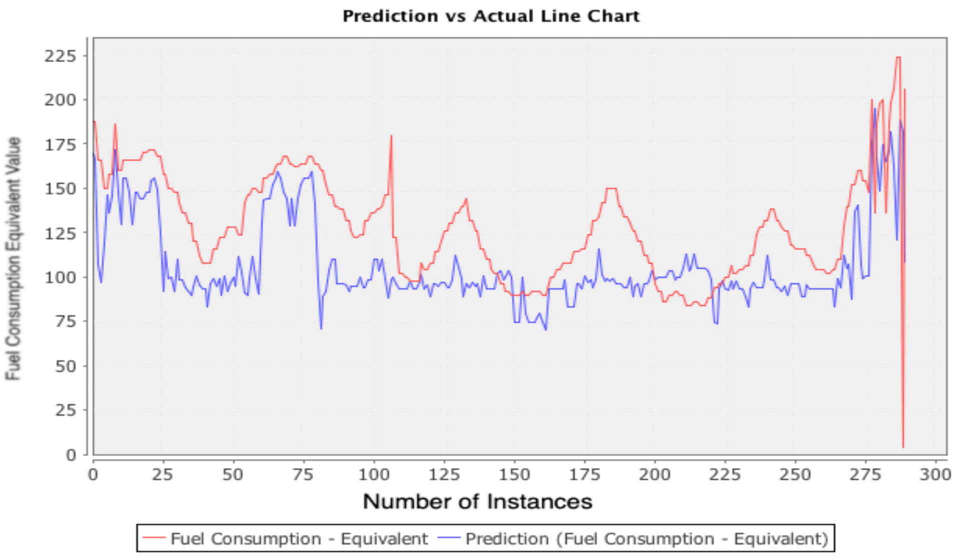

Figure 5.5 Case-2 - Values for Prediction vs Actual Fuel Consumption

5.4.3. Case-3

In Case-3, we inflated the tires 50 PS, and set cruise control speed to 80 km/h and open all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 50_80_Open_LOG.csv.

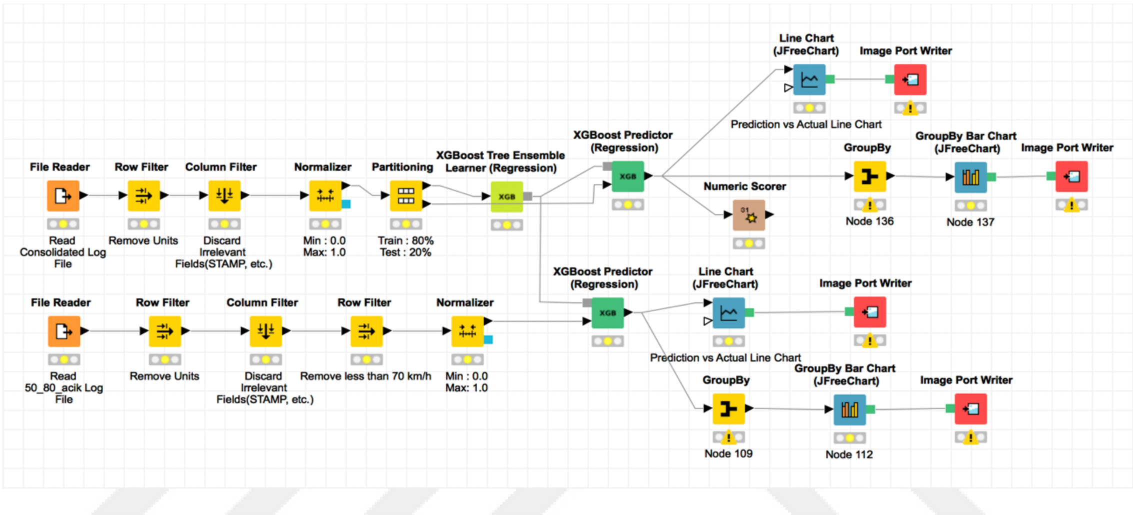

Figure 5.6 Model Representation of Case-3 on Knime

While the car dashboard showed the speed as 80 km/h, the CanBus gave the speed as 76 km/h. In addition to these, we decided to filter out records with speed less than 70 km/h to provide a realistic approach.

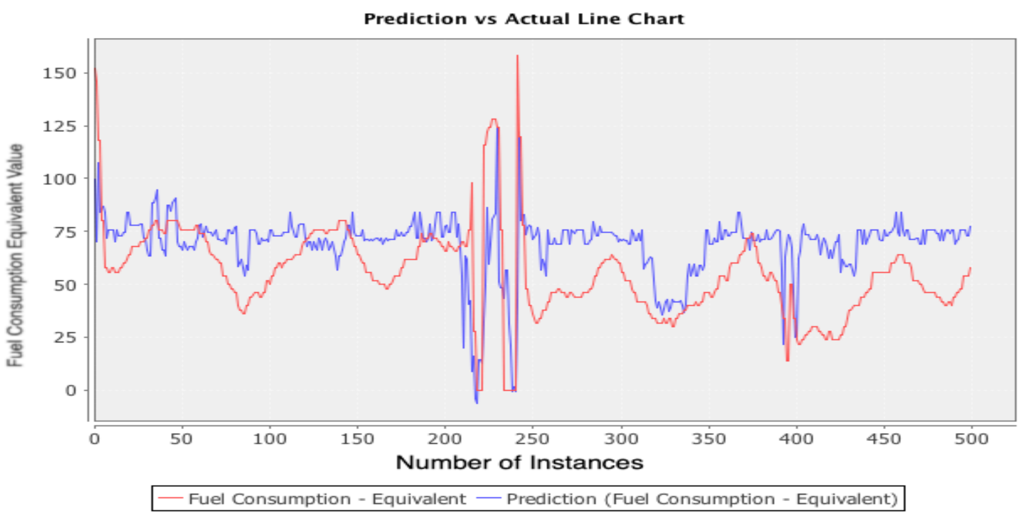

Figure 5.7 Case-3 - Values for Prediction vs Actual Fuel Consumption

5.4.4. Case-4

In Case-4, we continued with tire pressure as 50 PS, and set cruise control speed to 80 km/h and closed all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 50_80_Closed_LOG.csv.

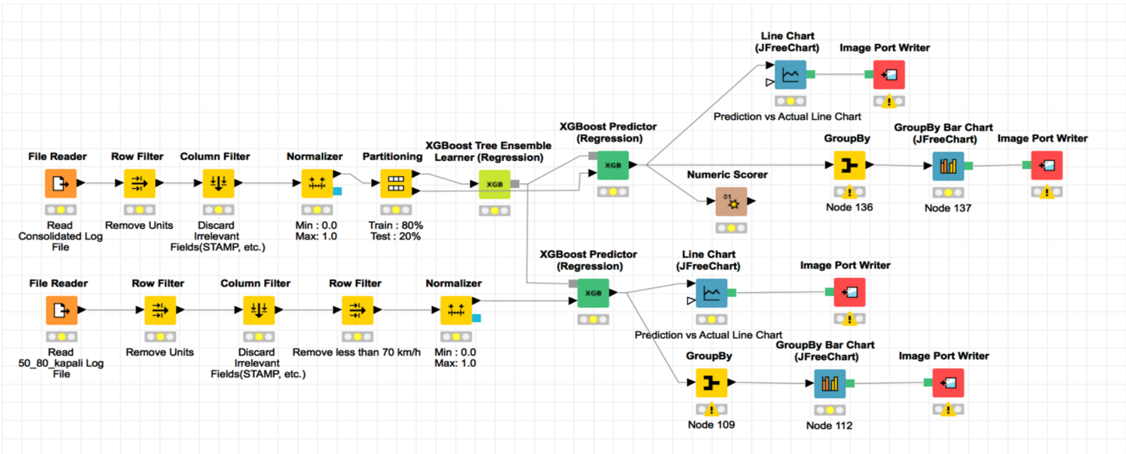

Figure 5.8 Model Representation of Case-4 on Knime

While the car dashboard showed the speed as 80 km/h, the CanBus gave the speed as 76 km/h. In addition to these, we decided to filter out records with speed less than 70 km/h to provide a realistic approach.

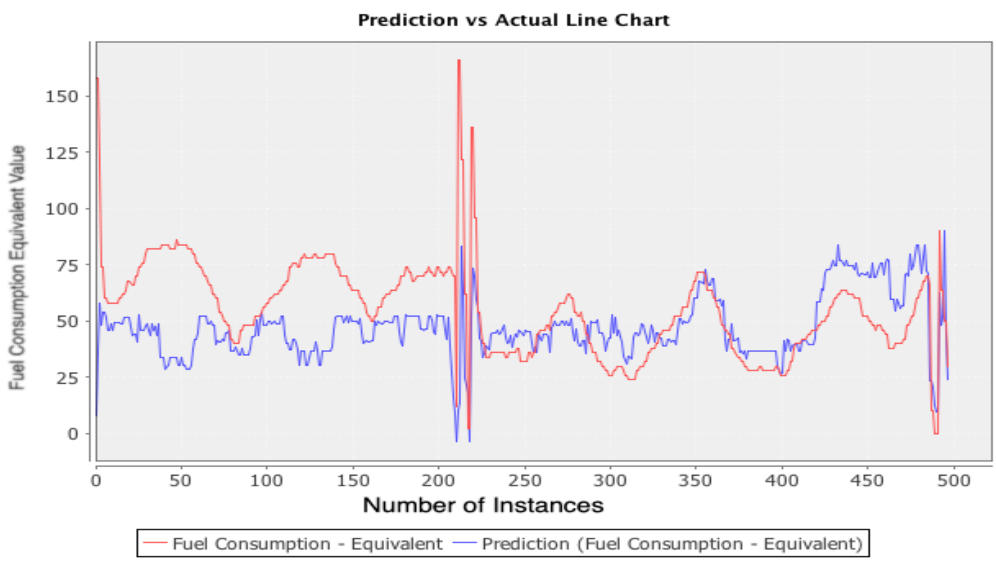

Figure 5.9 Case-4 - Values for Prediction vs Actual Fuel Consumption

5.4.5. Case-5

In Case-5, we continued with tire pressure as 50 PS, and set cruise control speed to 120 km/h and open all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 50_120_Open_LOG.csv.

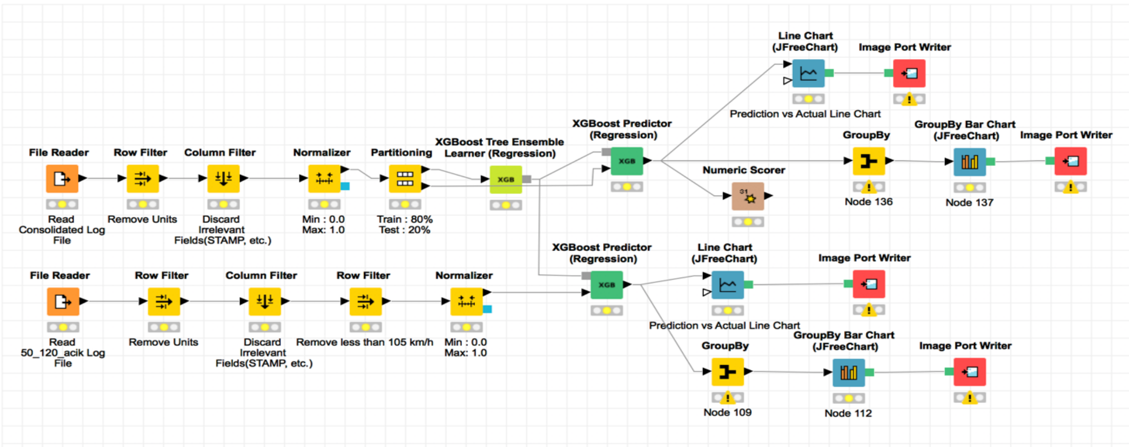

Figure 5.10 Model Representation of Case-5 on Knime

While the car dashboard showed the speed as 120 km/h, the CanBus gave the speed as 114 km/h. In addition to these, we decided to filter out records with speed less than 105 km/h to provide a realistic approach.

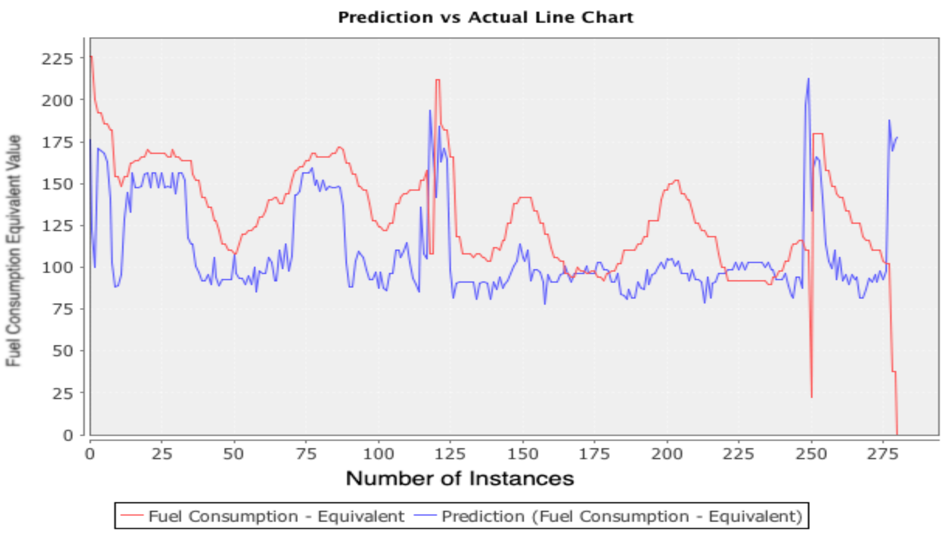

Figure 5.11 Case-5 - Values for Prediction vs Actual Fuel Consumption

5.4.6. Case-6

In Case-6, we continued with tire pressure as 50 PS, and set cruise control speed to 120 km/h and closed all windows. We drove our test car with Manual mode in DSG and used 6th gear. We logged our all trip data in a csv file as 50_120_Closed_Gear6_LOG.csv.

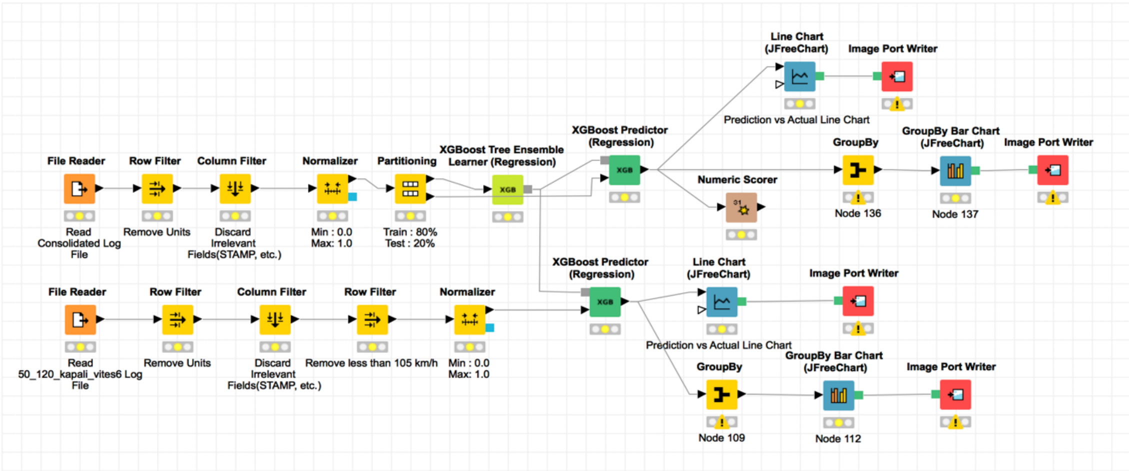

Figure 5.12 Model Representation of Case-6 on Knime

While the car dashboard showed the speed as 120 km/h, the CanBus gave the speed as 114 km/h. In addition to these, we decided to filter out records with speed less than 105 km/h to provide a realistic approach.

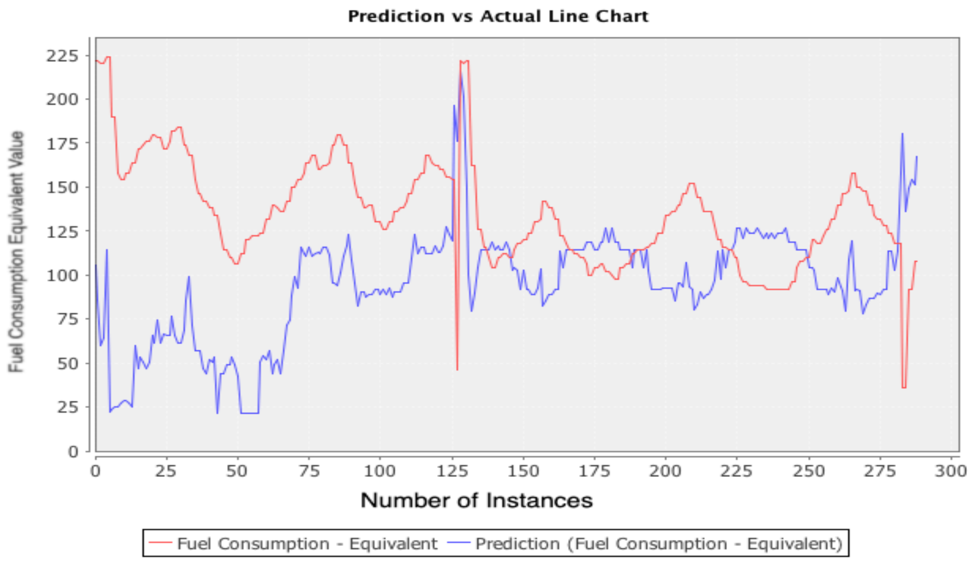

Figure 5.13 Case-6 - Values for Prediction vs Actual Fuel Consumption

5.4.7. Case-7

In Case-7, we continued with tire pressure as 50 PS, and set cruise control speed to 120km/h and closed all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 50_120_Closed_LOG.csv.

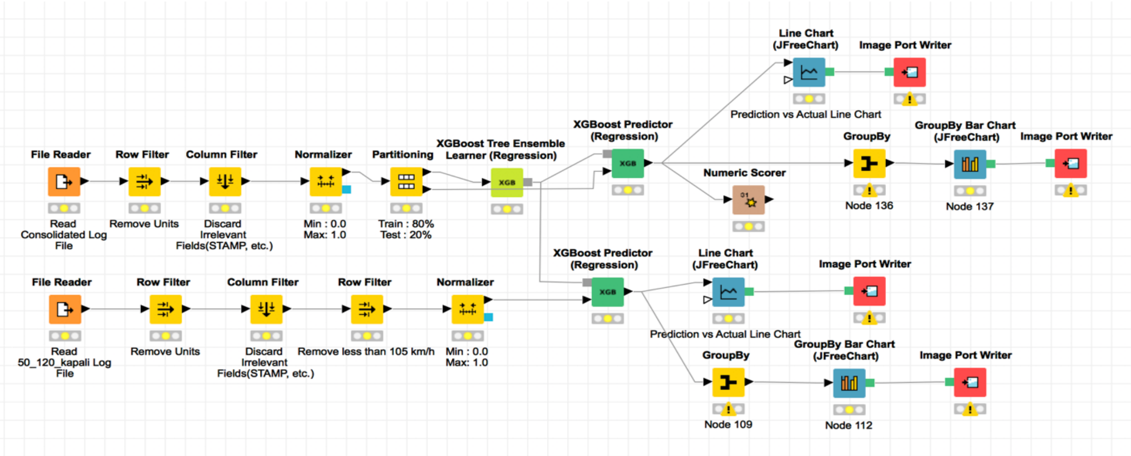

Figure 5.14 Model Representation of Case-7 on Knime

While the car dashboard showed the speed as 120 km/h, the CanBus gave the speed as 114 km/h. In addition to these, we decided to filter out records with speed less than 105 km/h to provide a realistic approach.

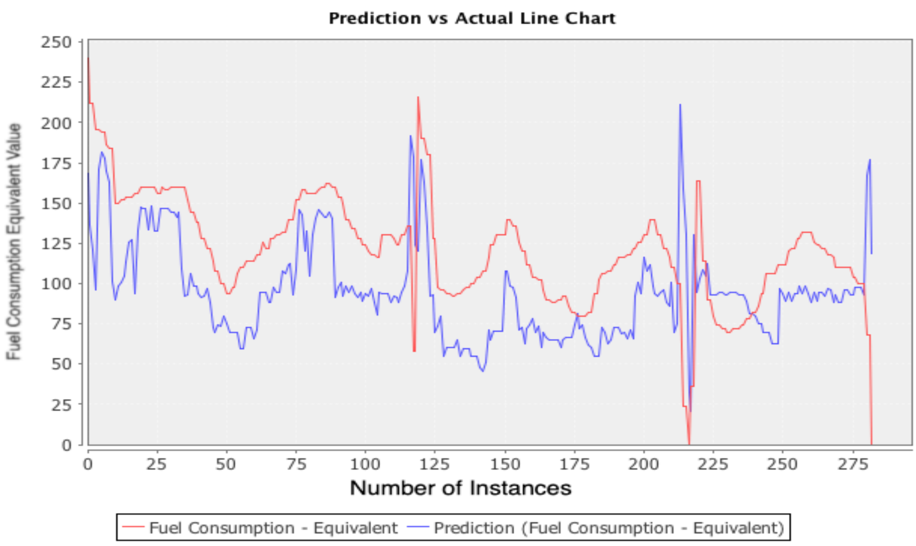

Figure 5.15 Case-7 - Values for Prediction vs Actual Fuel Consumption

5.4.8. Case-8

In Case-8, we continued with tire pressure as 50 PS, and set cruise control speed to 160 km/h and closed all windows. We drove our test car with Drive mode in DSG. We logged our all trip data in a csv file as 50_160_Closed_LOG.csv.

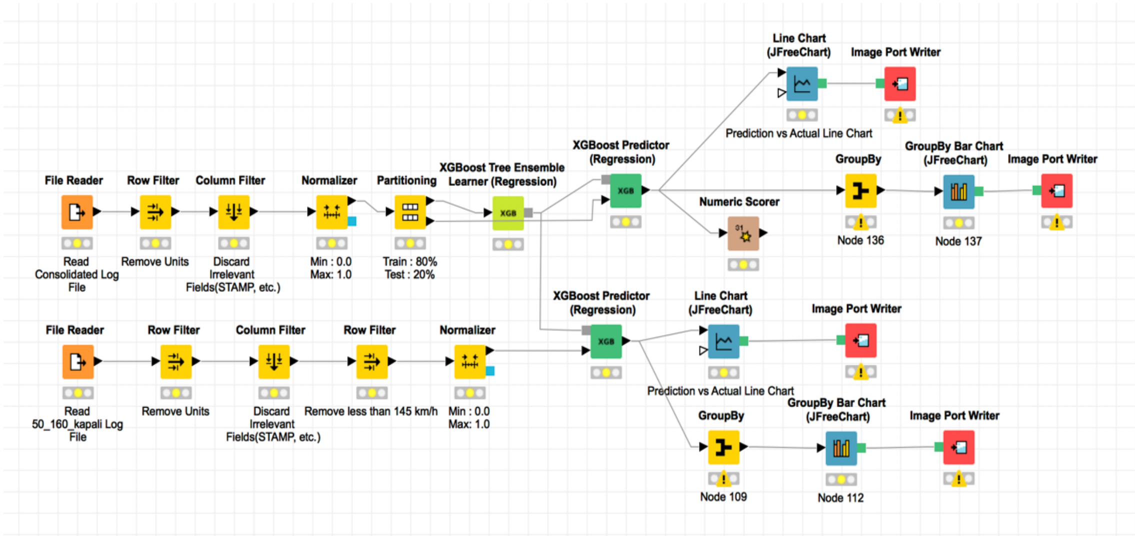

Figure 5.16 Model Representation of Case-8 on Knime

While the car dashboard showed the speed as 160 km/h, the CanBus gave the speed as 114 km/h. In addition to these, we decided to filter out records with speed less than 145 km/h to provide a realistic approach.

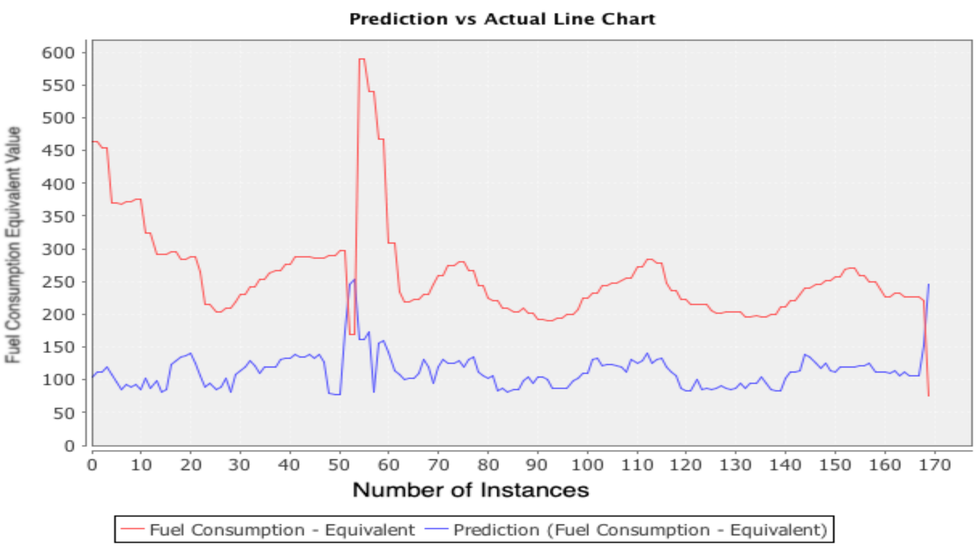

Figure 5.17 Case-8 - First 500 Values for Prediction vs Actual Fuel Consumption

5.5. Test Results

Table 5.3 Prediction and Actual Fuel Consumption Mean Values for All Cases

Test Cases	*Actual Fuel Consumption*	*Predicted Fuel Consumption*
50_80_Closed	55.462	41.442
50_80_Open	56.375	49.841
22_80_Closed	64.579	44.353
50_120_Closed (Case-7)	120.629	95.626
22_120_Closed	128.476	94.954
50_120_Open	131.473	91.1
50_120_Closed_Gear6	135.253	83.883
50_160_Closed	260.847	96.74

We have already tested the algorithm performance in terms of accuracy. According to the results, the model worked very well and produced minor errors on average. As we stated previous sections, the trained model did not cover all of the test cases. In this case, the model could not give meaningful results. We assumed that we can scale all our prediction value by using our base configuration's predicted value. The predicted value is 25.003 units lower than the actual value. If we accept that our algorithm calculates the consumption nearly equal, we can say that our model generally produces results lower and we need to scale it by adding this difference to all predicted value. Scaled prediction values can be shown in Table 5.4.

Table 5.4 Scaled Prediction Mean Values and Results

Test Cases	*Actual Fuel Consumption*	*Scaled Predicted Fuel Consumption*	*Modification Results*
50_80_Closed	55.462	66.445	POSITIVE
50_80_Open	56.375	74.844	POSITIVE
22_80_Closed	64.579	69.356	POSITIVE
50_120_Closed (Case-7)	120.629	120.629	Base Case
22_120_Closed	128.476	119.957	NEGATIVE
50_120_Open	131.473	116.103	NEGATIVE
50_120_Closed_Gear6	135.253	108.886	NEGATIVE
50_160_Closed	260.847	121.743	NEGATIVE

Our solution points that if a predicted value is lower than the actual value, the modification for this case is now working well and the car started to consume more than before. In this case, we indicate that the modification result is NEGATIVE. Otherwise, we highlighted it as POSITIVE. According to the test results, the Table 5.4 is obtained. The table shows that the car under 50_80_Closed, 50_80_Open and 22_80_Closed configurations consumes less than the stock version. Other cases, the configurations were no use to decrease fuel consumption.

6. CONCLUSION AND FUTURE WORKS

High fuel consumption negatively affects financial situations of drivers. Moreover, harmful gases such as carbon dioxide, carbon monoxide, ozone and peroxyacetyl nitrate are released into the atmosphere by burning of the fuel and cause serious air pollution.

In this thesis, generating a machine learning model of the fuel consumption from various vehicle sensor data and using the produced model to compare the effects of modifications on the vehicle with the previous setup were aimed.

The procedure was started by collecting sensor data from the vehicle. Collected data was tested with three machine learning algorithms. These algorithms were ANN, Gradient Boosted Trees Algorithm and XGBoost. These algorithms were run on the Knime tool. As a result of the tests, it was seen that the most appropriate algorithm for collected data was XGBoost. Fuel consumption model was created with XGBoost algorithm.

After the fuel consumption model created, various modifications were made on the vehicle and the results were compared with the model. These modifications included tire pressure changes (22ps, 32ps and 50ps), vehicle speed changes (80km/h, 120 km/h and 160 km/h) and vehicle window position changes (completely opened and completely closed).

All tests data are experimented on the built model to analyse the effect of modifications on the vehicle fuel consumption. In this study, we made an assumption that a coherent offset is added to predicted fuel consumption value to obtain a more practical model. The test results show that the test vehicle consumes less under 50_80_Closed, 50_80_Open and 22_80_Closed configuration set. On the other hand, the test vehicle consumed more under 22_120_Closed, 50_120_Open, 50_120_Closed_Gear6 and 50_160_Closed configuration set.

It is planned to improve the generated model by collecting much more data at different conditions (different weather conditions, different routes). The performance of the model also can be improved by changing the data sampling rate. It is also possible to add other vehicle sensor and phone GPS data to the collected data. Various machine learning algorithms can be tried for generating the fuel consumption model. In the future, various test cases can be experimented such as with different tire brands and types(winter/summer/all-season), with more and less load and so.

In addition to these; algorithm model should be trained in reference to the car's factory specs to compare properly.

All these processes will serve to produce vehicles with more efficient and low fuel consumption. With the reduction of fuel consumption, consumer's cost for fuel will be decreased. Moreover, reducing vehicle fuel consumption will decrease air pollution.

REFERENCES

Çapraz, A. G., Özel, P., Şevkli, M. & Beyca, Ö. F., 2016. Fuel Consumption Models Applied to Automobiles Using Real-Time Data: A Comparison of Statistical Models. Procedia Computer Science, Cilt 8, p. 774 – 781.

Alessandrini, A., Filippi, F. & Ortenzi, F., 2009. Consumption calculation of vehicles using OBD data. Rome, CTL, Centre For Transport and Logistics, University of Rome.

Almèr, H., 2015. Machine learning and statistical analysis in fuel consumption prediction for heavy vehicles, Stockholm: Kth Royal Instıtute Of Technology School of Computer Scıence And Communıcatıon.

Alpaydin, E., 2016. Introduction to Machine Learning. 2 ed. London: The MIT Press.

Brownlee, J., 2016. What is Deep Learning?. [Online]
Available at: https://machinelearningmastery.com/what-is-deep-learning/
[Accessed March 2019].

Chen, T. & Guestrin, C., 2016. XGBoost: A Scalable Tree Boosting System. New Tork, arXiv.

Cipollone, R., Battista, D. D. & Mauriello, M., 2015. Effects of oil warm up acceleration on the fuel consumption of reciprocating internal combustion engines.

L'Aquila, ATI 2015 - 70th Conference of the ATI Engineering Association.

Corrigan, S., 2016. Introduction to the Controller Area Network (CAN). [Online]
Available at: http://www.ti.com/lit/an/sloa101b/sloa101b.pdf
[Accessed March 2019].

Dude, R., 2018. What Is the OBD-II Port and What Is It Used For?. [Online]
Available at: https://www.makeuseof.com/tag/obd-ii-port-used/
[Accessed March 2019].

Du, Y., Wu, J., Yang, S. & Zhou, . L., 2017. Predicting vehicle fuel consumption patterns using floating vehicle data. Journal of Environmental Sciences, September, Cilt 59, pp. 24-29.

Gurney, K., 2004. Neural networks—an overview. %1 içindeAn introduction to neural networks. London: UCL Press, p. 13.

Hunting, B., 2017. Understanding Engine RPM Basics. [Online]
Available at: http://knowhow.napaonline.com/understanding-engine-rpm-basics/
[Accessed March 2019].

Kean, A. J., Harley, R. A. & Kendall, G. R., 2003. Effects of Vehicle Speed and Engine Load on Motor Vehicle Emissions. California, American Chemical Society.

Korolev, M. & Ruegg, K., 2015. Gradient Boosted Trees to Predict Store Sales, California: Stanford University.

Mitchell, T., 1997. Machine Learning. United States: McGraw-Hill.
Neriya, R., K, P. & K, G., 2017. Vehicle Health Monitoring and Analysis. Tamil Nadu, TIFAC CORE for Automotive Infotronics.

Perrotta, F., Parry, T. & Neves, L. C., 2017. Application of Machine Learning for Fuel Consumption Modelling of Trucks. Applications of Big Data in the Transport Industry, pp. 11-14.

Puget, J. F., 2016. What Is Machine Learning?. [Online]
Available at:
https://www.ibm.com/developerworks/community/blogs/jfp/entry/What_Is_Machine_Learning?lang=en
[Accessed March 2019].

Rahman, A. & Smith, A. D., 2017. Predicting Fuel Consumption for Commercial Buildings with Machine Learning Algorithms. Energy and Buildings.

Setiawan, R., 2018. Understanding a car air intake system. [Online]
Available at: https://www.sgcarmart.com/news/writeup.php?AID=127
[Accessed March 2019].

Wickramanayake, S. & Bandara, H. D., 2016. Fuel Consumption Prediction of Fleet Vehicles Using Machine Leaning: A Comparative Study. Moratuwa, Sri Lanka, IEEE.

Zeng, W., Miwa, T. & Morikawa, T., 2015. Exploring Trip Fuel Consumption by Machine Learning from GPS and CAN Bus Data. Eastern Asia Society for Transportation Studies, Volume 11.

BIOGRAPHICAL SKETCH

Cem Yeniçeri was born on March 29, 1990, in Gaziantep. After graduating from Gaziantep Anatolian High School in 2008, he began to study in Electricals and Electronics Engineering department in Eskisehir Osmangazi University. He graduated in 2013 and after that, he enrolled in the M.Sc. program in Computer Engineering department at Galatasaray University.

Outside of academics, he has worked as a software engineer for various IT companies which are Savronik, Huawei, Central Securities Depository of Turkey and eBay Turkey | Gittigidiyor.com chronologically for six years. Then, he established his own company Unicraft Software Solutions in United Kingdom.

He published a conference paper for his bachelor thesis, entitled as “A Smart Solution for Transmitter Localization” supervised by Assoc. Prof. Dr Ahmet Yazıcı was presented at the “International Symposium on INnovations in Intelligent SysTems and Applications (INISTA), 2013, Bulgaria” conference.