AUTOMATED WEIGH-IN-MOTION THROUGH VEHICULAR TELEMATICS AND MACHINE LEARNING

by

SIVARAMALINGAM KIRUSHANTH

Dissertation submitted in fulfilment of the requirements for the degree

Doctor of Philosophy: Informatics

in the **Faculty of Informatics and Design**

at the **Cape Peninsula University of Technology**

Supervisor: Dr. BONIFACE KABASO

Cape Town

August 2019

DECLARATION

I, Sivaramalingam Kirushanth, declare that the contents of this dissertation represent my own unaided work and that the dissertation has not previously been submitted for academic examination towards any qualification. Furthermore, it represents my own opinions and not necessarily those of the Cape Peninsula University of Technology.

10.03.2020

Signed Date

ABSTRACT

Road safety is one of the major concerns in today's world. Driving an overloaded vehicle causes various ill-effects to road safety. Various kinds of Weigh-in-motion (WIM) systems are used to control and reduce the impacts of overloaded driving. Existing WIM systems are either expensive or slower and influenced by various factors. Advancement in network connectivity and sensor devices led to the development of the Internet of Vehicles (IoV), a subfield of the growing Internet of Things (IoT). IoV is powered by Vehicular Telematics (VT), also known as flying car data. VT data is used by the transport industries for many reasons such as fleet management, insurance (pay as you drive), driving behaviour detection, and road anomaly detection. Intelligent Transportation System (ITS) uses both IoV and Machine Learning (ML) techniques to build an automated Artificially Intelligent (AI) transportation system.

According to the Newtonians' physics and literature, under certain conditions, the driving force needed by a vehicle to obtain a particular acceleration is influenced by the total weight of the vehicle. That implies that if driving force and other influencing parameters are known, we could infer the weight of a vehicle. VT data can be used to obtain many features, including the driving force. This dissertation discusses the effort taken to validate the idea of inferring the weight of a vehicle using VT and ML. This research involved designing and testing the prototype artefact. The Design Science Research (DSR) methodology was used in this research. The C-K design theory was used in this DSR. The application of C-K theory in DSR has shown the different dimension for approaching applied research. A pragmatist approach was used in the design and development of this research.

According to the C-K design theory, with all the knowledge, K0, from literature and the laws of physics, we formed an initial concept C0: "A new WIM solution using VT and ML", with the propositions p1: "faster", p2: "economical/cheaper", p3: "Ubiquitous". The concept was tested by designing and developing the prototype (artefact). A backend to process VT data using ML was developed as a by-product of this research. We have tested several ML algorithms during the development stage, and an Artificial neural network (ANN) architecture of three hidden layers with 30 nodes in each layer has shown astounding performance with Accuracy = 0.945, R-Squared = 0.97, Adjusted R-Squared = 0.97, Mean Squared Error = 34.68, Residual Standard Error = 6.03. The ANN outperformed all other tested ML algorithms on the collected VT dataset. We can infer the weight using the smaller dataset obtained from the context of a small car. Results from small cars show the supports for the concept theory.

•

ACKNOWLEDGEMENTS

Thanks to **Nature** for moulding/holding me and for making me always wonder.

Many thanks to my supervisor **Dr. Boniface Kabaso**, for his patience, dedication, support, motivation, and everything from the very first day onwards.

I share my gratitude to my dearest sister **Mrs. Sujitharan Subajini**, my brother **Mr. Sivaramalingam Vijayashanth**, my brother-in-law **Mr. Gopalasingam Sujitharan** and my friends **Mr. Gnanapragasam Naveendrakumar** and **Mr. Manickavasagar Kayanan,** for their unconditional support.

I am very grateful to the staff of the Vavuniya Campus of the University of Jaffna as well as the staff and colleagues at the Department of Information Technology, Faculty of Informatics and Design, Cape Peninsula University of Technology.

DEDICATION

I wish to dedicate this dissertation to my beloved **father** late **Mr. A. K. Rajasingham Sivaramalingam,** and to my loving **mother Mrs. Sumathy Sivaramalingam.** I am nothing without them.

Publications from this Research

- Kirushanth, S. and Kabaso, B., 2018, July. Telematics and Road Safety. In *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)* (pp. 103-108). IEEE.

- Kirushnath, S. and Kabaso, B., 2018, July. Weigh-in-motion Using Machine Learning and Telematics. In *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)* (pp. 115-120). IEEE.

- Kirushanth, S. and Kabaso, B.,2019, October. Designing a Cloud-native Weigh-In-Motion, In 2019 Open Innovations Conference (pp. 25-29). IEEE.

- Kirushanth, S. and Kabaso, B. (2020) 'Design and Development of Weigh-In-Motion Using Vehicular Telematics', *Journal of Sensors*, 2020, pp. 1–22. doi: 10.1155/2020/7871215

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

AI: Artificial Intelligence is the area which concerns about making the machines to be intelligent.

ANN: Artificial Neural Network is a computational network that is one of the ML techniques inspired by biological neurons.

FCD: Floating Car Data is the data from vehicles about their location and speed.

ECU: Engine Control Unit is the electronic controlling unit of a vehicle's sensors and actuators

ELM327: ELM327 is a programmed microcontroller produced by ELM Electronics for translating the on-board diagnostics (OBD) interface found in most modern cars

GNSS: Global Navigation Satellite System is the common name for all the navigation systems which uses positioning satellites

GPS: Global Positioning System is one of the GNSS available for public use.

GVWR: Gross Vehicle Weight Rating is the maximum operating mass/weight of a vehicle as specified by the manufacturer.

GVM: Gross Vehicle Mass is the same as GVWR.

HS-WIM: High-Speed Weigh-in-motion is used to weigh the vehicles at high speeds.

IMU: Inertial Measurement Unit is one of the sensor units comprising Accelerometer, Gyroscope, and Magnetometer.

IoT: Internet of Things is all about the remote sensors, remote actuators, and communication systems that work over the internet.

IoV: Internet of Vehicles is a sub-domain of IoT, focused on things related to vehicles.

ITS: Intelligent Transport System is the domain concerns about the amalgamation of Artificial intelligence in Transportation Systems

JSON: JavaScript Object Notation is a data-interchange format that is easily readable to humans, and lightweight.

LoRaWAN: Long Range Wide Area Network

LS-WIM: Low-Speed Weight-In-Motion

ML: Machine Learning is all about making the machines to perform without using explicit instructions.

MLP: Multilayer Perceptron is a type of feedforward Artificial neural network

OBD: On-Board Diagnostic provides an interface to access ECU data.

PAYD: Pay as You Drive an Insurance scheme.

UBI: Usage-Based Insurance.

WIM: Weigh-in-motion is the way of weighing a vehicle in motion.

CLARIFICATION OF TERMS

Mass: The mass of an object is a fundamental property of the object; a numerical measure of its inertia; a fundamental measure of the amount of matter in the object. The usual symbol for mass is m, and its SI unit is the kilogram. While the mass is typically considered to be an unchanging property of an object, at speeds approaching the speed of light, one must consider the increase in the relativistic mass.

Weight: The weight of an object is the force of gravity on the object and may be defined as the mass times the acceleration of gravity, $w = mg$. Since the weight is a force, its SI unit is the Newton (N). The SI measure kg for mass is used in general (assuming the gravitational acceleration is constant at any point on the earth's surface).

Acceleration: Acceleration is the rate of change in velocity (dv/dt). Acceleration is defined as the rate of change of velocity. Acceleration is inherently a vector quantity, and an object will have non-zero acceleration if its speed and/or direction is changing. The units for acceleration can be implied from the definition to be meters/second divided by seconds, usually written m/s^2.

In this research, we assume the gravitational acceleration g is constant ($9.8ms^{-2}$) in all the places where the data was collected such that the weight of a mass is constant. Thus, finding Mass or Weight refers to the same objective.

CHAPTER 1
INTRODUCTION

This dissertation discusses the development of a new WIM solution using VT and ML. This research was initiated after performing a Systematic Literature Review (SLR) on VT and Road Safety. This problem was identified as one of the findings of that SLR. The detailed description of the SLR is discussed in Chapter 2. This chapter discusses the introduction to the research by briefly discussing the following:

- Section 1.1. How concerning is road safety in the global dimension and in South Africa?
- Section 1.2. What is overloading and how does it affect road safety?
- Section 1.3. What are the available solutions for WIMs and their Pros and Cons?
- Section 1.4. What is VT, and its current usage?
- Section 1.5. Problem Statement of this research.
- Section 1.6. Aim and Objectives
- Section 1.7. Methodology
- Section 1.8. Dissertation Structure

1.1 Driving and Road Safety

Driving and road safety are current and growing problems with global dimensions (Meiring and Myburgh, 2015). According to the comprehensive status report on road safety conducted by the World Health Organisation (WHO), 1.24 million traffic-related fatalities occur annually worldwide (WHO, 2015). Notably, in 2011, South Africa had the highest number of fatalities, according to the International Road Traffic and Accident Database (IRTAD) annual road safety report 2013 (Meiring and Myburgh, 2015). Due to the amplified necessity for mobility in developed and developing countries, the growth in vehicle manufacturing is inevitable. Driver assistance and safety awareness programmes have been some areas of focus to minimise road safety incidents, and since the WHO launched their “Decade of Action for Road Safety (2011–2020)” programme, a remarkable improvement in road safety has been noticeable. According to the U.S. Department of Transportation, it was observed that two factors, namely vehicle condition and road/environment conditions, were collectively responsible for 5.2% of the road accidents in the U.S (Magaña and Muñoz-Organero, 2017).

The contribution of human behaviour towards traffic accidents is an essential area of interest in the remedial attempts to address the global road safety problem (Meiring and Myburgh, 2015). Risk-taking driving behaviour plays a significant role in most of the accidents. It is discussed that over speeding, sudden acceleration/deceleration hard cornering, and not wearing seatbelts are some of the risk-taking driving behaviours (Wahlström, Skog and Händel, 2015).

1.2 Overloading

Kerb Weight is the total weight of a vehicle with a full tank of fuel and excluding accessories, luggage, and passengers. Gross Vehicle Weight (GVW) is the maximum allowed weight of a vehicle when fully loaded. GVW is the sum of Kerb Weight and Payload. The payload is the maximum load a vehicle can carry as specified by the manufacturer. Vehicles loaded with more than the payload or that weigh more than GVM are considered as overloaded (Oastler, 2015).

Driving an overloaded vehicle is an illegal and punishable offence in most of the countries as it leads to accidents and infrastructural damages (Haugen *et al.*, 2016; Lydon *et al.*, 2016). South African National Road Traffic Regulations state the overloading scenarios which lead to prosecution for an offence under regulations in the National Road Traffic Act, 1996 (Act No. 93 of 1996).

Overloaded vehicles pose severe threats to road transport operations. Increased risks for road users, severe impacts on the durability of infrastructure, especially for bridges and pavements, and on fair competition between operators are some of the major threats by overloaded vehicles (Jacob and Véronique, 2010).

1.2.1 Effects of Overloading

Jacob & Feypell-de La Beaumelle (2010a) listed the following as the negative impacts of overloading:

- ***Accident risk and accident severity:***

The likelihood of an overloaded vehicle involved in an accident is higher than the non-overloaded or legally loaded vehicle. The consequences of such accidents are more severe (Tolouei, Maher and Titheridge, 2013). Momentum is defined by the velocity into the mass. Impact of collision is proportional to the mass for a given velocity. In other words, the higher the mass of a vehicle at a certain speed in a direction (velocity), the higher its impact due to its higher kinetic energy. Vehicles which are specially designed to carry heavy loads have a certain operational speed limit; for example, according to the Motor Traffic Department of Sri Lanka, all heavy vehicles must drive at the maximum speed of 40 km per hour regardless of the road and time. This is to prevent the severity of impact by reducing the velocity of a greater mass.

- ***Vehicle instability:***

The stability of an overloaded vehicle is questionable due to the increased centre of gravity (mass). Objects with a higher centre of gravity are more prone to topple or rollover. The increased inertia with the higher centre of gravity may lead to lane departure or knife-jacking.

- ***Braking default:***

The design of the braking system of vehicles allows a maximum weight specified by the manufacturer. The braking system of any vehicle is meant for the most allowable weight specified by the manufacturer. The breaking capability depends on the brakes themselves, however conjointly on the tire and suspension performances designed for the most allowable weight of the vehicle. Any weight in excess reduces the braking capability of a truck and will even damage the braking system.

- ***Loss of motivity and manoeuvrability:***

An overloaded vehicle becomes under-powered; this results in lower speeds on up-hill slopes as well as the risk of congestion, inefficient engine braking and over-speeding on down-hill slopes. Overtaking also takes longer, and thus incurs additional risks for the other road users.

- ***Tire blow-outs:***

Tire blowouts can occur due to the induced overheating of tires due to overloads. The severity of such an event is higher when flammable and toxic goods are transported.

- ***Damage to the infrastructure:***

Apart from the threats to road safety, the overloaded vehicles also increase pavement wear. It also causes bridge damage, especially on older bridges. There were reports on bridge collapse due to overloaded vehicles.

- ***Economic impact:***

Overloading prompts substantial mutilations in cargo transport rivalry, between transport modes (for example rail, waterborne and street), and between street transport organizations and administrators. In France, it was assessed that a 5-pivot enunciated truck, worked at 20% over-burden lasting through the year, created an extra 25 000 € advantage for each year (Jacob and Véronique, 2010). Overloading likewise implies an infringement of the tax collection rules, for example, vehicle enlistment charges, axle duties, and toll framework expenses. It is accordingly essential to authorize vehicle weight and measurement guidelines to limit the quantity of overloading on larger than average trucks. The improvement of cutting-edge truckload checking frameworks, either ready or out and about, as a feature of Intelligent Transportation Systems (ITS), offers significant potential and elective answers for conventional roadside implementation by consistence officers.

In summary, overloaded vehicle cause various ill effects such as a vehicle's mechanical component degradation (Anthony, 2013), air pollution by increased greenhouse gas emissions (Wahyudi, Ganis and Taufik Mulyono, 2014), increased fuel consumption (CWCSA, 2017), and road infrastructural damages (Huang, Zhang and Yi, 2009; Pais, Amorim and Minhoto, 2013). Furthermore, an overloaded vehicle becomes less stable as the centre of mass changes. This leads to less traction control and difficulty in steering. In addition to that, since it needs extra braking distance, an overloaded vehicle is more prone to road hazards(CSIR (Roads and Transport Technology), 1997). Additionally, an overloaded vehicle becomes a cause of traffic congestion and causes risks when overtaking as it goes underpowered (Shah *et al.*, 2016).

Reducing the number of overloaded vehicles is likely to reduce the number of crashes (Jacob and Véronique, 2010). However, it is the responsibility of governments, vehicle manufacturers, researchers and road users to reduce the number of fatalities. "*The Department of Transport, in conjunction with provincial traffic authorities, the South African National Roads Agency Limited (SANRAL) and the Council for Scientific and Industrial Research (CSIR) has drafted the National Overload Strategy to address the problem of overloaded vehicles. The strategy covers the issues of self-regulation by the freight industry, funding, training and operational issues and a review of the 5% tolerance on the mass limit that is allowed for in the Road Traffic Act.*"(Jonck, 2017).

1.3 Weigh-in-motion

Determining the weight of a vehicle, also known as WIM, has been done in various ways. WIM is a useful tool to contribute to more compliance with mass regulation. It has been used most successfully for nearly two decades. WIM has helped to reduce the number of overloaded vehicles and contributed to the more efficient and effective use of police officers' time. A reduction in overloaded trucks is also conducive to a reduction in crashes (Jacob and Véronique, 2010).

1.3.1 Impact of WIM Systems

In their paper, Haugen et al. (2016) listed seven significant impacts of WIM systems, which are:

- **Carrier behaviour change**

WIM technologies allow the control of vehicle weights without disrupting the traffic and freight operations. In some countries like France and the Netherlands, the B-WIM system comes with video surveillance. The system enters the weight violated vehicle information to the centralised remote database. This has proved to have a positive impact on the loading behaviour by the carrier (Stanczyk et al., 2012 and Jacob et al., 2010).

- **Protection of road pavement infrastructure**

Highly loaded vehicles cause damages to road constructions and pavements. They affect the basic infrastructure. The WIM systems can be used to reduce the cost of the resurfacing and repair works of these structures caused by severely overweight vehicles. Several studies done in the United States of America (USA), revealed experiences in tracking large pavement damages caused by heavy loaded vehicles, from the reports provided on weekly, monthly and yearly basis. This methodology was applied by carrying out an evaluation process and comparing the damage prior to the installation of the WIM system and the damage after the installation of weight enforcement. The estimation of the damage is done through equivalent single axle loads (ESAL) factors; a concept which was developed at the American Association of State Highway Officials (AASHO) monthly calculations were performed to get an estimate of the pavement damage attributed to the excess weight of vehicles at each site. This system contributed to a large extent in reducing the cost of damages in road pavement and infrastructure; especially in Norway, where the government took high interest in this, particularly in the places where there are no other means to control weight in main motorways.

- **Traffic Safety**

A vehicle even loaded with safety precautions, will be dangerous and risky for the stability and durability of the vehicle. This will result in brake system faults, handling and difficulty in control. The risk of an accident is higher in vehicles that are overloaded compared to the vehicles which are fairly loaded. Furthermore, as the number of vehicles involved in the routine traffic increases, it increases the severity of the accidental consequences. The research done by Jacob et al. (2010) revealed that there is a lack of statistical data related to accidents caused by overloaded vehicles and data were not being collected by police. Several studies have found that highly loaded vehicle drivers are suspected of using other routes to get rid of weigh-in scales. This causes a rise in accidents on these secondary roads. This is the reason why secondary roads are not open to overloaded vehicles in Norway. Due to the implementation of the WIM system in Norway on all the infrastructures, a number of accidents and incidents have been lessened.

- **Traffic management**

The WIM system may help in providing an outline of overloaded vehicles and their geographical location in regional and national views to traffic management centres. In Norway, the data from this WIM system could be relevant for 5 main regions. In management of overloaded vehicles, accurate, reliable and updated information of routine traffic, vehicle weight and its category are useful to manage traffic volume, lane usage and speed. In future WIM data analysis may be used in newly introduced applications that are developed for traffic control centres that deal with real online traffic management, dangerous goods transportation, heavy vehicle flow, and safety measurements in tunnel traffic.

- **Freight Planning**

The WIM system data can be used in the management and planning of large freight transportation. In Norway, a short message is delivered to each overweight vehicle owner and freight service proprietor. A yearly report would give them an outline of their company's efficacy and performance in terms of loading vehicles which are overweight. This will provide the facility to arrange their goods to load in order over their vehicles. Furthermore, this will definitely affect the freight operations by reducing the time spent on weight controls and increase efficiency. Hence reducing vehicle overloading will produce safer transportation, which is safer to all. This could be assessed by using questionnaires and collection of data and analysis conducted among the drivers and other employees.

- **Environmental impacts**

Assessing the effects on the environment due to overloading with noise, vehicle vibrations and air pollution is complicated. Measurement of these impacts should be implemented as soon as possible. Specially designed equipment should be used to measure only the vehicle-borne impacts as there are other means of vibrations such as ground-borne vibrations. At the same time, other than overloading, there are other parameters that influence gas emissions, such as driving pattern and road gradient. They are normally measured by factors that are related to vehicle category, emission from the engine, the total weight of the vehicle and speed (Poulikakos, 2010).

- **Economic benefits**

Overloaded vehicles produce benefits in a negative manner for carriers by violating the regulations for taxes and other payments (Jacob and Véronique, 2010). The authorities of the WIM system will benefit only in a few ways such as penalties, taxes and fees. But the efficiency of weight control system reduces the overall costs born to repair the infrastructural damages, resurfacing works and road/tunnel closures. It will definitely be useful to calculate the total cost reduction compared to previous years that are caused by overweight vehicles at regional and national level, through a benefit-cost study in Norway.

1.3.2 Summary of WIM Systems

Traditional WIM uses stationary scales such as static-weighbridges which are commonly used to measure industrial and commercial vehicles such as trucks and lorries (Nichols and Bullock, 2004). They, however, take quite a long time to weigh each vehicle. Moreover, their costs of system installation and maintenance are expensive (Barsanescu, Carlescu and Stefanescu, 2007). Static-weighbridges are more accurate than non-static weighbridges, only when the vehicle is stationary. Non-static WIMs (low-speed/high-speed WIMs) are deployed inroads to detect load violations. Non-static WIMs use various parameters to detect a vehicle's weight (Kim *et al.*, 2009). Some of the parameters of non-static WIMs are pavement vibration (Bajwa *et al.*, 2013), and magnetic signal based on single micro-electro-mechanical system (MEMS) magnetic sensor (Lan *et al.*, 2011).

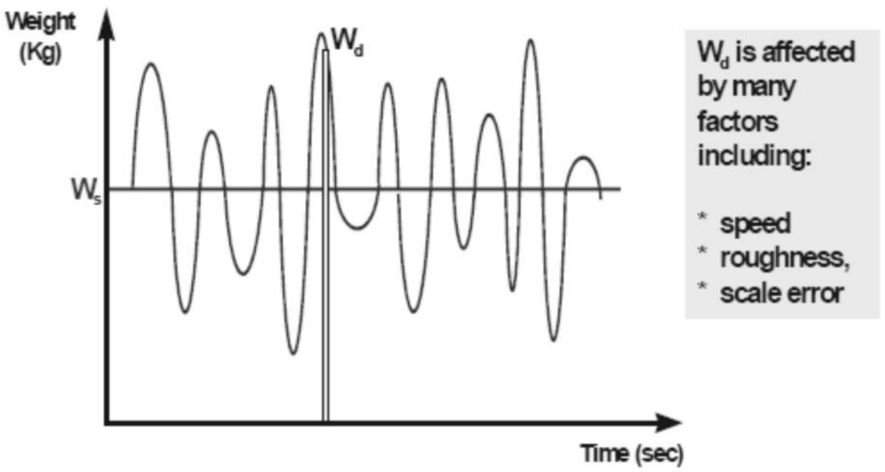


Figure 1.1: Fluctuation in the WIM reading due to internal and external factors (Bushman & Pratt, 1998)

Figure 1.1 by Bushman & Pratt (1998) shows the variations in reading the weight of a vehicle over time using Static and Dynamic WIM systems.

Counting the number of goods and passengers by visual inspections is so common that an automatic onboard passenger count detection system was developed and tested successfully (Xu and Zhao, 2011). However, since each passenger has a different weight and goods weigh differently, the visual inspections do not provide correct results all the time (CWCSA, 2017).

Emerging technologies such as smart tyres (Pirelli, 2017) with the aid of a tyre pressure monitoring system (TPMS) are used in WIM (Matsuzaki and Todoroki, 2008). However, such systems are more expensive as they require the pressure sensors to be installed on every tire. For infrastructural enforcement, several practices of the WIM system are being used all around the world. High-speed WIM systems are used in countries like Taiwan with large tolerance levels of up to 30%. This tolerance level of up to 30% is acceptable since there are vehicles in high frequency. In some countries use portable WIM systems which are used to detect overloads in a very short period of time. But the accuracy of portable WIM systems is low compared to the conventional one and the efficiency or pre-selection is low (Jacob and Véronique, 2010). Table 1.1 shows the pros and cons of the existing WIM systems.

Table 1.1: Pros and Cons of Existing WIM Systems

WIM Systems	Pros	Cons
Conventional (Static-Weighbridge)	More Accurate	Time-consuming, vehicles need to be stationary
Non-static weighbridges (using ML)	Faster than traditional, Deployed on the roads	Less accurate than conventional needs more calculations, influenced by external factors
Goods and Passenger count	Easy and fast in an ideal condition	Inaccurate in most cases
Tyre Pressure (Smart Tyres/using ML)	Faster	Expensive

The latest applications of WIM systems are to be introduced in traffic regulation and overload vehicle control. In the near future, these WIM systems have the probability of using onboard WIM systems. Thus, each and every vehicle could be monitored in terms of their weight and impact infrastructure. Another study suggests that real-time overload monitoring could reduce the damage cost in large scale according to their data transmission system and GPS which are necessary to eliminate overload vehicles (Jacob and Véronique, 2010).

1.4 Vehicular Telematics

VT has been widely used in vehicle tracking, fleet management and insurance industries for more than a decade (Tong *et al.*, 2016; Wahlstrom, Skog and Handel, 2017). According to Fleming (2010: 6) “Telematics is an unstoppable trend in cars”. ML approaches are widely used in identifying driving behaviour and road anomaly detection (Wahlstrom, Skog and Handel, 2017). Currently, ML approaches are only used in WIM systems using weighbridges and TPMS based systems. So far, the advent of VT has made it possible to obtain information on the attributes of a vehicle even while it is in motion, since ML is known to support inference making from both static and dynamic data.

Developing a WIM system using VT and ML would benefit the transportation industry by reducing time, cost, and errors compared to other WIM systems. The proposed WIM system was expected to be fast, easy, accurate, and less expensive. Transportation and enforcement industries would be able to monitor the overloaded vehicles with the proposed solution remotely. This dissertation explains the effort taken to integrate the application of VT and ML for WIM in a way that will improve on existing approaches concerning cost, accuracy, and speed.

1.5 Background to the research problem

Driving overloaded vehicles is one of the significant causes of road hazards (Karim *et al.*, 2014). Vehicle overloading causes road accidents due to loss of control, brake and tyre failure, and uneven wear and tear. Vehicle overloading also causes road infrastructural damages and excessive fuel consumption, which leads to environmental pollution (Wahyudi, Ganis and Taufik Mulyono, 2014). Overloading is a common problem in transportation.

Finding the overloaded vehicle on any road segment is a challenging task. Counting the number of goods and heads is commonly found in many places. Non-static weighbridges on roads and static weighbridges are used in WIM. Seat and chassis mounted weight scales, using tyre pressure, and smart tyres are the new solutions available in WIM (Xu and Zhao, 2011; Anthony, 2013; Jonck, 2017; Pirelli, 2017). Visual inspection and counting do not always give an accurate result, and it is impossible for a vehicle in motion. Using mass scales in weighbridges are costly and time-consuming solutions. Chassis and seat-mounted scales use mechanical devices which need frequent calibration; further, the reading varies during the drive. Smart tyres and measuring weight using tyre pressure is expensive since it needs to be installed on every wheel.

1.6 Problem Statement

Driving overloaded vehicle causes road infrastructural damages, accidents, air pollution by excessive fuel consumption, and unexpected expenses. Measuring the gross weight of a vehicle on a road segment without interrupting the traffic flow is a problem worth researching, and its solutions have several economic benefits.

Currently, vehicles must drive through time-consuming weighbridges to measure weight. Dynamic weighbridges are only available on specific road segments. Additionally, the tyre pressure monitoring system (TPMS) solutions, which give parameters to measure weight, are more expensive to install (Matsuzaki and Todoroki, 2008). Other conventional weighing apparatus used are less accurate while on the move and need frequent calibration (Anthony, 2013).

“There are still issues and challenges for WIM technology and application which require more research and development work. It is also essential to better disseminate knowledge and best practices, to exchange experiences, and carry out large scale common tests of WIM sensors and systems” (Jacob and Véronique, 2010). It is imperative to research and develop a system that infers the weight of a vehicle in a fast, reliable and non-intrusive way using VT and ML.

1.7 Aim and Objectives

This research aims to apply ML and VT in WIM to aid the transportation industry.

Objectives of this research are to:

1. Identify the relevant development platforms, parameters (features), and algorithms to infer the weight of a vehicle in motion.
2. Design a conceptual framework that integrates VT and ML for WIM.
3. Develop a prototype system that leverages VT and ML to determine the weight of a vehicle in motion.
4. Evaluate the prototype system in terms of performance (accuracy, speed), usability and cost.

1.8 Research Area

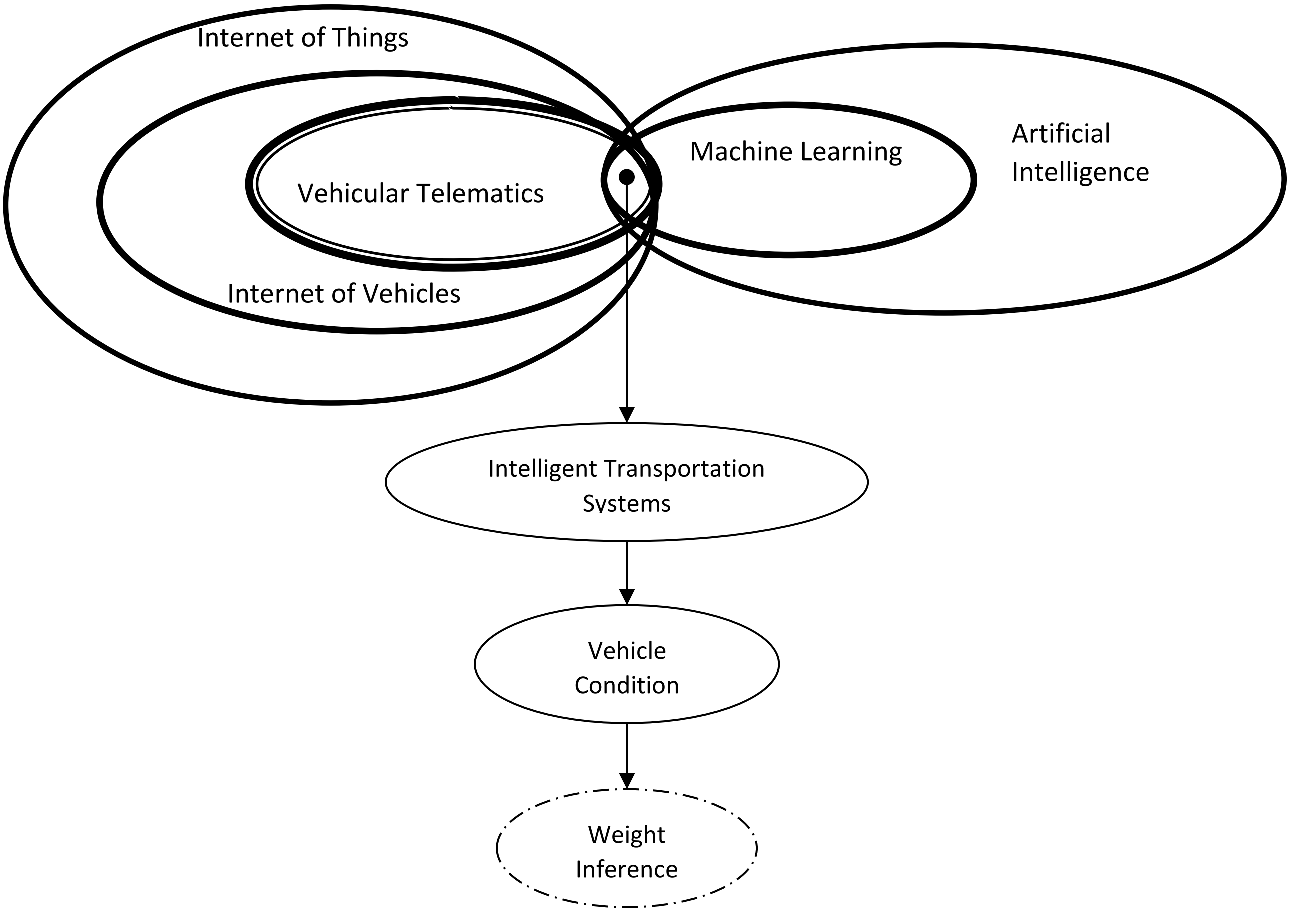


Figure 1.2: Research Area

Figure 1.2 shows the research area of this project. This research falls within two major areas, namely Artificial Intelligence (AI) and the Internet of Things (IoT). AI is powered with ML. IoV is a subfield of IoT. IoV is powered with VT. ML and VT power intelligent Transportation Systems. Vehicle condition monitoring is one of the research fields in ITS. This research focused on the Weight inference part of the Vehicle condition monitoring in ITS.

1.9 Methodology

1.9.1 Philosophy

This research comfortably fits under the assumptions of pragmatism. According to Camarinha (2012:40) "*assumption of pragmatism is not committed to any one system of philosophy or reality. Truth is what works at the time. We need to stop asking questions about reality and the laws of nature and start solving problems*". We can also look at this philosophy in between assumption of the post (Positivism) and assumption of Interpretivism. Positivism is a way of making claims first and then testing, refining, or abandoning some of them for other claims more strongly justified. In assumptions of Interpretivism, researchers seek to understand the context and then make an interpretation of what they find, which is shaped by their own experiences and backgrounds (Camarinha-matos, 2012). Herbert E. Simon is the founding father of design science research (DSR). Well known for his research on AI, economics and decision making (Hevner *et al.*, 2010), his works reveal that researchers in AI fit under the DSR category. Henver (2007) classified DSR under the pragmatic category of research. "*Pragmatism is a school of thought that considers practical consequences or real effects to be vital components of both meaning and truth. Along these lines, I contend that design science research is essentially pragmatic in nature due to its emphasis on relevance; making a clear contribution into the application environment."* (Hevner, 2007). DSRs have different metaphysical assumptions. Shift on ontological and epistemological viewpoints take place in circumscription cycles depicted in Figure 3.3 (Vaishnavi and Kuechler, 2004).

DSR follows the pragmatist approach. Artefacts were developed and evaluated; the evaluation used quantitative results from practical experiments to obtain a single truth, which is the weight of a vehicle is always singular. Since this is a pragmatic type of research, this research study followed a combination of inductive and deductive approaches.

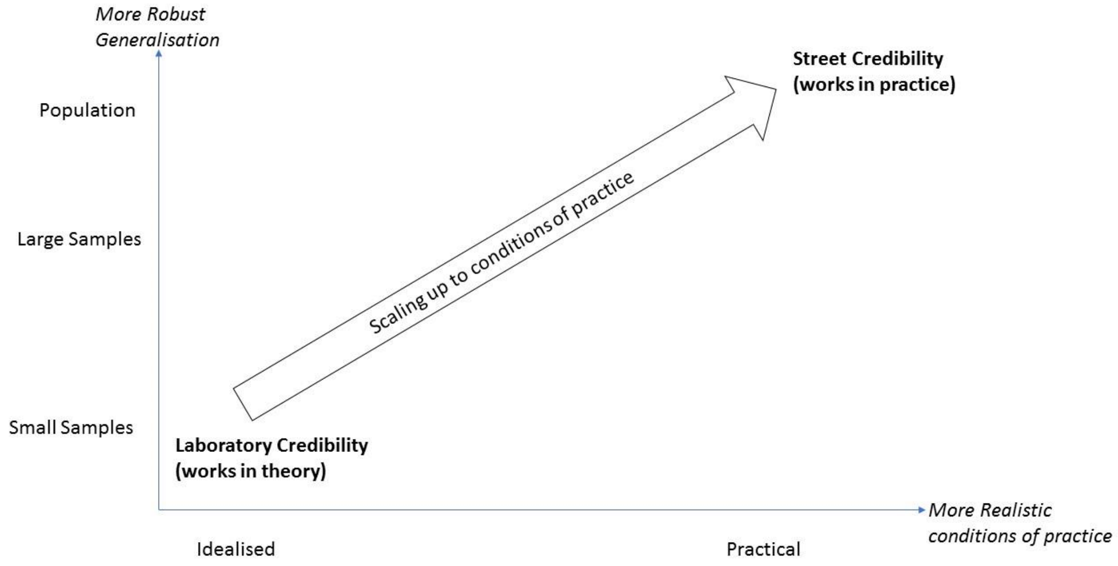


Figure 1.3: Idea to Practice (Wieringa, 2014)

This research was carried out with Wieringa's (2014) scaling up from lab to practice research strategy where the developed prototype WIM system was evaluated for its performance on a small scale within a context of a car, but with the potential to test with a large dataset in future.

1.10 Vehicle weight inference system

Figure 1.4 shows an overview of the proposed vehicle inference system. At the beginning of the research, it was planned to present a solution as a black-box system connected to the cloud. The black-box has to collect sufficient data from ECU and GNSS. The geolocation, along with the time stamp has to be sent to the GIS database (weather source). The GIS database may return the weather and road condition information to the weight inference system. The weight inference system should collect all the inputs from the black-box and weather sources and infer the weight using a trained ML model.

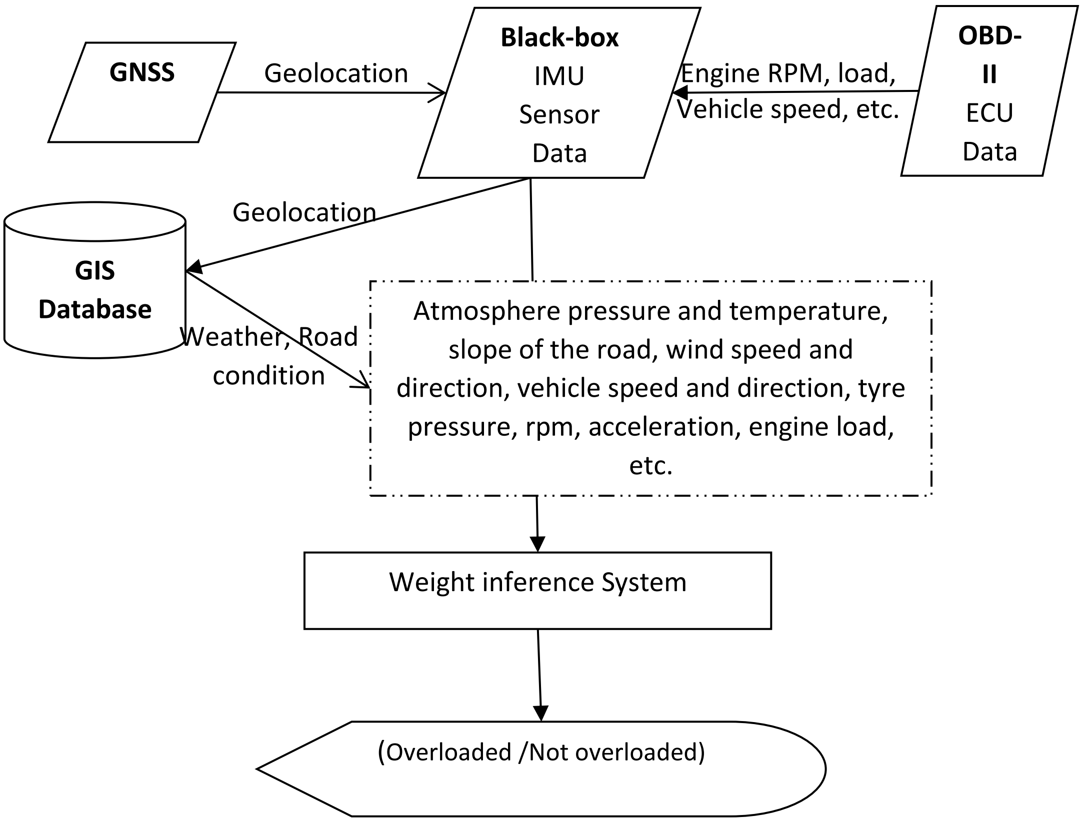


Figure 1.4: Overview of the Proposed Vehicle Weight Inference System approach

The proposed system was developed as a prototype system consisting of a smartphone and an OBD-II Bluetooth Scanner as the "Black-box". The weight inference system named "WIM application" was developed. The development of the prototype design is explained in Chapter 4.

1.11 Conceptual Framework

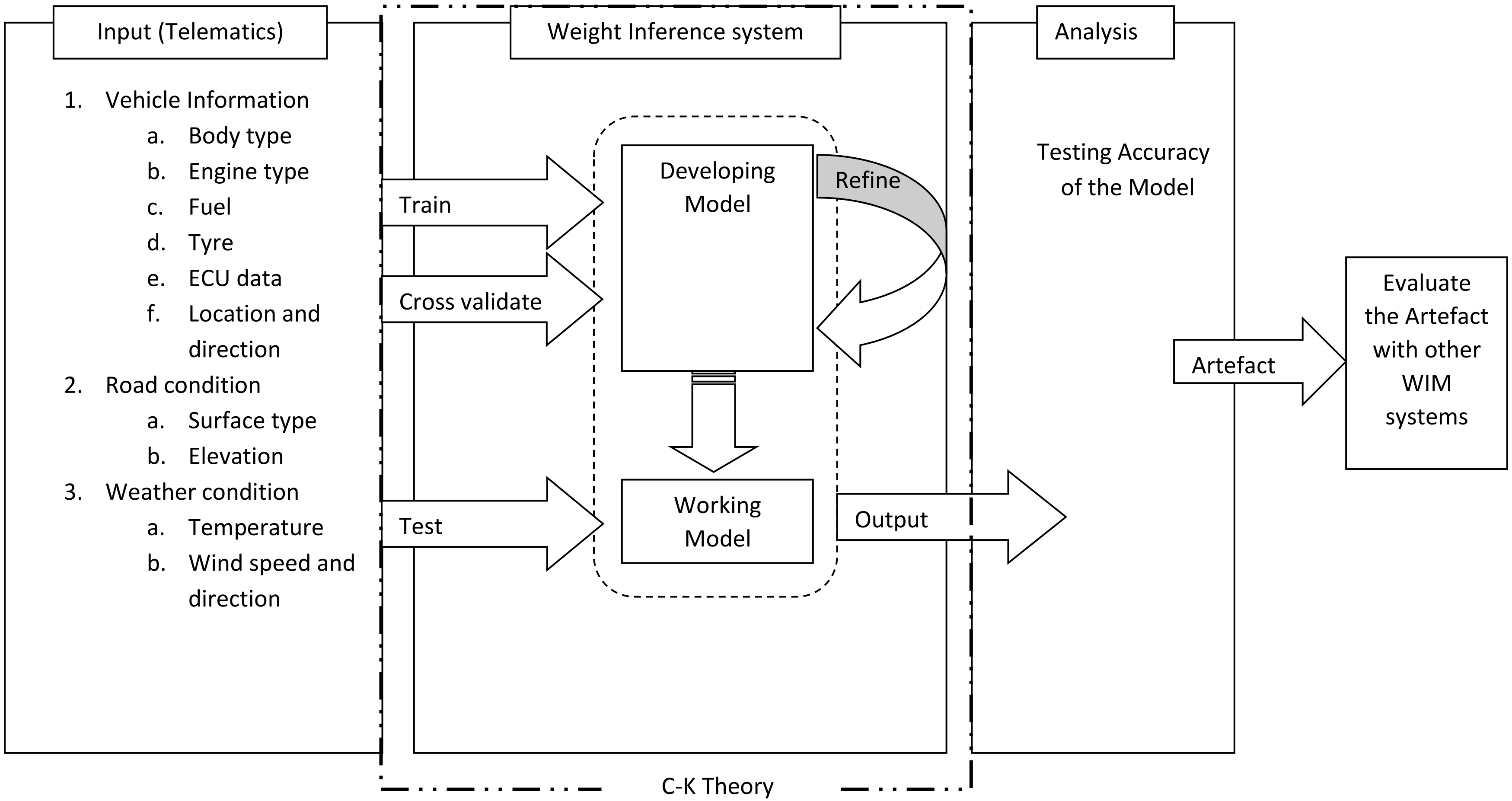


Figure 1.5: Conceptual Framework of the research

Figure 1.5 describes the conceptual framework of the research. This research can be viewed in four phases. Phase 1 includes the collection of enough telematics data and other information. This focuses on choosing appropriate sources and methods of data collection, storing, and transmission. Phase 2 involves the use of appropriate ML algorithms, building the model, training, cross-validating, and testing the model. Phase 3 involves validating the prototype by testing on controlled and uncontrolled environments. Phase 4 is evaluating the system with existing WIM solutions. Phase 1 & 2 can be viewed under the development stage of DSR, and Phase 3 & 4 fits under the Evaluation phase of DSR. A unified design theory, Concept-Knowledge (C-K) theory (Hatchuel and Weil, 2002), was used during phases 1, 2 & 3.

1.12 Data Collection

Three different datasets were collected in this research. One is in the Development (Phase 1 &2) and two are in the Evaluation stages (in Phases 3 and 4) in this DSR.

1.12.1 Data for Development Stage (Phase 1&2)

Vehicle information such as vehicle model, kerb weight, GWM, fuel type, engine capacity, transmission type, and wheel information (such as radius, width) was collected from the vehicle records. ECU information was collected from the OBD-II device. Location information was obtained using GNSS on mobile phones. Weather information of a specific geo-location was collected from reliable and freely available OpenWeatherMap API. Known payload weights were used during the training and testing phase. These data were used to build and test ML models.

1.12.2 Data for Evaluation Stage

Phase 3

The output from the Development stage is an ML model; this ML model was tested on-site using known weights. The correctness and the accuracy of the inference system were evaluated at this stage.

Phase 4

The system (prototype) was compared against some parameters such as correctness, cost, and speed with other WIM systems. The parameters were collected from the literature.

1.13 Evaluation

A designed artefact must be evaluated on the utility, quality, and efficacy of it. This evaluation must be demonstrated via well-executed evaluation methods (Hevner and Chatterjee, 2010). The ML models are the most critical artefact in this research. ML models return abstract values for further computation. The inference model performance was compared with R-squared, adjusted R-squared, and accuracy.

1.14 Delineation of the research

This research focused on ground vehicles, mainly cars and multipurpose vehicles, which are manufactured after the year 2000 (with OBD-II interface). The OBD device was only used for data collection from ECU, not to write into the ECU.

1.15 The significance of the research

This research may enable the production of a system to remotely view the weight of a vehicle in motion in a fast and non-intrusive way. The system would be beneficial for the transport industries and traffic enforcement departments.

1.16 Expected outcomes, results and contributions of the research

It was expected to provide a non-intrusive solution to existing problems in WIM systems. The solution is a black-box and a cloud-based ML system (back-end). As mentioned in C-K design theory, the final knowledge C_K is from unsuccessful and successful experiences of implementation of the design and associated tools. Findings were communicated by publishing in accredited journals and conferences. The practical contribution of this research is made by introducing a new approach in WIM system technology, new variables (features), and comparing ML models. The theoretical contribution of this research was that we theorised that the VT data could be used to infer the weight of a vehicle. The methodological contribution of this research was the application of C-K design theory in this design science research. The practical contribution of this research was the implementation of the proposed WIM approach by developing a prototype.

1.17 Ethical consideration

This research collected data from the vehicle, including the geographical position and speed by installing a probing device on the vehicle. The data collected was stored as blind data (without vehicle identity and driver information).

1.18 Dissertation Structure

This dissertation comprises six chapters. **Chapter 1** provides an overview of the research.

Chapter 2 discusses the background to the research and discusses the research Objective 1, which is identifying the relevant development platforms, parameters (features), and algorithms to infer the weight of a vehicle in motion.

Chapter 3 discusses the philosophical stance and research methodology.

Chapter 4 explains the research Objective 2 & 3, which is the design and development of a conceptual framework and a prototype system that leverages VT and ML to determine the weight of a vehicle in motion. Selection of features and different ML algorithms are discussed in this chapter.

Chapter 5 discusses the results obtained using the different ML algorithms used in Chapter 4 and research Objective 4, which is the evaluation of the prototype system in terms of performance (accuracy, speed), usability and cost.

Chapter 6 concludes the dissertation by briefly discussing the research objectives and future directions.

CHAPTER 2

BACKGROUND

This chapter discusses some important factors which are essential to this research. This chapter is organised as follows:

Section 2.1. Systematic Literature Review on VT and Road Safety

Section 2.2. Detailed History of WIM Systems

Section 2.3. Introduction to Onboard Diagnostics Module

Section 2.4. Theoretical Background to this Research

2.1 Systematic Literature Review on VT and Road Safety

This section presents the paper *'Telematics and Road Safety'* (Kirushanth and Kabaso, 2018) a Systematic Literature Review (SLR) compiled to find the use of VT in Road safety. This SLR has shed more light on finding the research gap in order to commence this research.

2.1.1 Introduction

Road safety is one of the major concerns all around the world. Notably, in 2011, South Africa had the highest number of fatalities, according to the International Road Traffic and Accident Database (IRTAD) annual road safety report 2013 (Meiring and Myburgh, 2015). Governments, vehicle manufacturers, and other stakeholders are involved continuously in ensuring road safety through several means. VT also known as flying car data (FCD), is one of the technological solutions available to ensure road safety. Telematics data comprising the geolocation of the vehicle, speed, acceleration, engine control unit information, and some other data are used by some vehicle insurance and fleet management companies. Use of telematic devices is becoming mandatory in some countries. It is believed that every car in the EU will be equipped with telematics sensors after the year 2018 (Braun, Reiter and Siddharthan, 2015). Research on the use of telematics data to detect driving behaviour and road anomalies show more significant success in vehicle fleet and road infrastructure industries (Meiring and Myburgh, 2015). Governmental and non-governmental organisations are collecting telematics data for various reasons, such as monitoring road usage and driving behaviour.

The intrusion of Usage-based Insurance (UBI) is a milestone in the use of telematics data, which introduced the Pay as You Drive (PAYD) scheme to attract customers. Risk-taking driving behaviour plays a significant role in most accidents. According to Wahlström, Skog and Händel (2015), over speeding, sudden acceleration/braking, hard cornering, and not wearing seatbelts are some of the risk-taking driving behaviours. Specially designed data collection devices (Black-Box) or smartphones were used to collect telematics vehicle data. The gathered telematics data includes information about vehicle movements and control inputs, from which it is possible to gather information about driving styles and behaviours. UBI using telematics devices often offer incentives and feedback on driving behaviour (Wahlström, Skog and Händel, 2015; Moosavi, Ramnath and Nandi, 2016; Tong *et al.*, 2016). Feedback is provided to the drivers or those who are responsible via in-vehicle data recorders (IVDR) (Toledo, Musicant and Lotan, 2008) or smartphones and other means of electronic communication such as text messages, emails and web sites. Several advanced driver assistant systems (ADAS) are available to assist drivers in preventing and reducing accidents, but it is only available on high-end model vehicles (Chaovalit, Saiprasert and Pholprasit, 2013). According to Lee (2007), there is an urgent need for researchers, designers, and policymakers to consider how to evict the causes of distraction and capitalise on the potential benefits of emerging technology. There is a need to research more cost-effective solutions available to assist drivers in reducing accident risks.

2.1.2 Background

In their review on UBI using telematics, Tselentis, Yannis and Vlahogianni (2017) discussed that most of the UBI applications use IVDRs such as On-Board-Diagnostics (OBD) modules to collect data, and smartphones to gather and transmit driving data to the central databases. Further, they firmly believed that smartphones would be mainly used for data acquisition in the future due to its high penetration rates in households as well as the high hardware cost of IVDRs. They also believed that drivers receiving feedback and monitoring driving behaviour would result in reducing crash risk.

Wahlstrom, Skog and Handel (2017) reviewed some notable academic and industrial studies with system aspects such as embedded and complementary sensors, energy-efficiency, and cloud computing. They discussed the methods to estimate smartphones' orientation and position with respect to some given vehicle frame. They also categorised smartphone-based driver classification based on sensors used, considered driving events, and applied classification techniques. Furthermore, they also reviewed road condition monitoring. They stated that while smartphones potentially can function as an enabler for low-cost implementations, there are often technical difficulties that must be overcome due to the non-dedicated character of the device. They concluded that future improvements in sensor technology would be beneficial for the road condition and driving detection, and that studies on communication standards and roadmaps need to be done in detail.

Meiring and Myburgh (2015) investigated various driving style analysis solutions. Their review focused more on the relevant ML and artificial intelligence (AI) algorithms utilised in driver behaviour analysis systems. They found that Fuzzy Logic inference systems, Hidden Markov Models (HMM) and Support Vector Machines (SVM) would provide promising results when on reduced model complexity. Tong et al. (2016) conducted a review of existing evidence of how vehicle telematics can affect accident rates, and how countries across the world have introduced policies regarding the use of telematics. Their main objective was to find evidence of the impact of telematics on accident risk, particularly in young and novice drivers. They found that no direct evidence shows telematics affects accident rates, but they found that parental involvement can indirectly influence young novice driver risk.

2.1.3 Aim and Objectives

The aim of this review is to find the use of telematics data in detecting driving behaviour, road anomalies, and finding the effect of feedback on driving behaviour.

The specific objectives of this research are to:

- Review existing evidence of the impact of telematics-based approaches for identifying driver behaviour.
- Review existing driver feedback techniques used to encourage safer driving.
- Identify how best telematics data can be accessed, compiled and used to produce the best driver feedback techniques to motivate good driving.

2.1.4 Methodology

2.1.4.1 Review Protocol

2.1.4.1.1 Research Questions

RQ1. What are the sources of vehicle telematics data?

RQ2. What are the sensors and features used to detect driving behaviour?

RQ3. What are the techniques used to detect or identify driving behaviour?

RQ4. What are the available driver feedback techniques?

RQ5. How does feedback affect driving behaviour?

2.1.4.2 Search Strategy

2.1.4.2.1 Search Terms

Telematic* AND Driving AND (On-Board OR In-Vehicle Data Recorder OR Smartphone) AND (Behaviour OR Pattern OR Style) AND Feedback AND (Safe OR Encourage) AND (Classification AND Machine Learning OR ML Algorithms)

2.1.4.2.2 Databases used

IEEE Xplore, ACM Digital Library, and Google Scholar.

2.1.4.3 Study Selection

2.1.4.3.1 Inclusion criteria:

- Empirical studies on driver behaviour detection using telematics data from onboard sensors and smartphones.
- Empirical studies on the driver feedback techniques and their effects.

2.1.4.3.2 Exclusion criteria:

- Studies before 1999 or not in English.
- Studies Using Driving Simulators.
- Review studies.

2.1.4.4 Quality Assessment Criteria

Table 2.1 shows the ten quality assessment (QA) questions used in this review process, adopted from Malhotra (2015). Scores are given by assigning 1 for strongly agree, 0.5 for partly, and 0 for disagree. Studies with scores less than 7.0 out of 10 for quality assessment question were rejected.

Table 2.1: Ten QA questions used in this SLR

Quality Assurance Questions
Are the aims of the research clearly stated?
Are the independent variables clearly defined?
Is the data set size appropriate?
Is the data-collection procedure clearly defined?
Is attributes sub-selection technique used?
Are the techniques clearly defined?
Are the results and findings clearly stated?
Are the limitations of the study specified?
Is the research methodology repeatable?
Does the study contribute/add to the literature?

2.1.4.5 *Data Extraction Form*

Table 2.2 shows the structure of the data extraction form. Data extraction form entries were populated by extracting the data from the studies filtered from QA criteria.

Table 2.2: Format of the Data extraction form

Variables
Authors:
Title:
Publication Type:
Year:
Field of Study:
Total Quality Assessment Score (out of 10):
Data collection devices (IVDR/Smartphone):
Sensors used (Accelerometer/Gyro/Speed/Geo Position/etc):
Features used (Acceleration/Speed/Cornering/etc):
Algorithms used (SVM/ANN/BN/etc) :
Feedback types used (Light/Audible/Haptic/Textural/etc):
Experiment Size:
Performance descriptor (Accuracy/F1 Score/ROC/AUC/etc):
Performance score (Accuracy /ROC value):
Strengths:
Weaknesses:

2.1.4.6 *Data Synthesis*

No statistical analysis has been made on the collected data. This paper contains some essential facts extracted from the selected studies, which are presented in textual and tabular forms. This report is furnished by summarising some selected literature which is selected based on the QA criteria.

2.1.5 Results and Discussion

Table 2.3 shows the number of studies at each stage of the review process.

Table 2.3: Review process results

Stage	**No. of Studies**	**Remarks**
Initial Search using search terms	4831	Using metadata
Filer by title	1258	
Filter by abstract	325	
Full texts gathered	274	No secondary studies
Included in Extraction form	90	High QA scores

Studies are summarised in a way to answer the research questions. Table 2.4 contains a list of studies used in this research.

Table 2.4: List of studies used in this SLR

Research Questions		Studies
RQ1	What are the sources of vehicle telematics data?	(Toledo, Musicant and Lotan, 2008; Dai *et al.*, 2010; Johnson and Trivedi, 2011; Castignani, Frank and Engel, 2013; RoSPA, 2013; Stoichkov, 2013; Meseguer *et al.*, 2013; Kalra, Chugh and Bansal, 2014; Pargaonkar *et al.*, 2014; Amarasinghe *et al.*, 2015; D. Chen *et al.*, 2015; Jarret Engelbrecht *et al.*, 2015; Khedkar and Ravi, 2015; Vlahogianni and Barmpounakis, 2017; Wahlstrom, Skog and Handel, 2017; Zabihi *et al.*, 2017)
RQ2	What are the sensors and features used to detect driving behaviour?	(Takeda *et al.*, 2012; Meseguer *et al.*, 2013; Wahlström, Skog and Händel, 2015; Vaiana *et al.*, 2014; Wahlstrom, Skog and Handel, 2014; Wahlström, Skog and Händel, 2014; Amarasinghe *et al.*, 2015; Castignani *et al.*, 2015b; D. Chen *et al.*, 2015; Ferreira *et al.*, 2017; Vlahogianni and Barmpounakis, 2017)
RQ3	What are the techniques used to detect or identify driving behaviour?	(Fazeen, M and Gozick, B and Dantu, R and Bhukhiya *et al.*, 2012; Meseguer *et al.*, 2013; Hong, Margines and Dey, 2014; D. Chen *et al.*, 2015; Z. Chen *et al.*, 2015; J. Engelbrecht *et al.*, 2015; Hosseinioun, Al-Osman and Saddik, 2016; Liu *et al.*, 2016; Ouyang *et al.*, 2016; Wu, Zhang and Dong, 2016; Saiprasert, Pholprasit and Thajchayapong, 2017)
RQ4	What are the available driver feedback techniques?	(Belz, Robinson and Casali, 1999; Tijerina *et al.*, 2000; Cummings *et al.*, 2007; Donmez, Boyle and Lee, 2007; Adell *et al.*, 2008; Toledo, Musicant and Lotan, 2008; Asif *et al.*, 2009; Farmer, Kirley and McCartt, 2010; Takeda *et al.*, 2012, 2011; Kruger, Hefer and Matthew, 2013; Simons-Morton *et al.*, 2013; Braun, Reiter and Siddharthan, 2015; Yan, Wang and Wu, 2016)
RQ5	How does feedback affect driving behaviour?	(Huang *et al.*, 2005; Lee, 2007; Dogan *et al.*, 2011; Simons-Morton *et al.*, 2013; Braun, Reiter and Siddharthan, 2015; Shimshoni *et al.*, 2015; Zabihi *et al.*, 2017)

2.1.5.1 What are the sources of vehicle telematics data?

There are two primary data collection methods used in most literature, namely: IVDRs and smartphones. Most of the UBI and fleet management application used IVDRs, and only a few of them used smartphones. There is much research done on the application of smartphones in telematics.

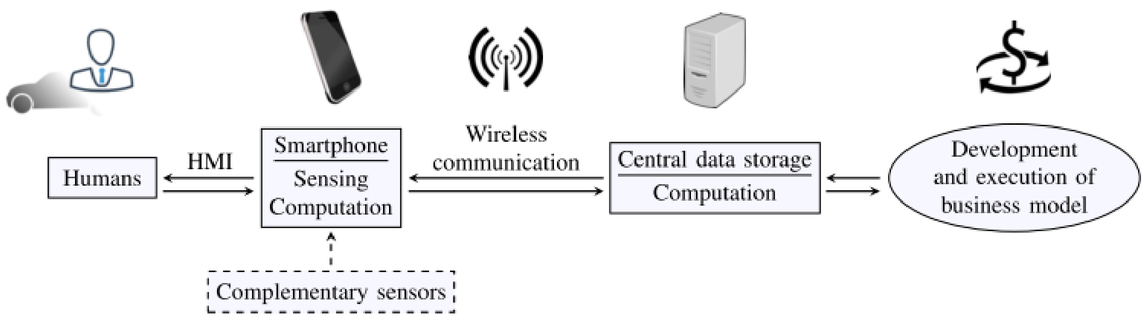


Figure 2.1: Process diagram illustrating the information flow of smartphone-based vehicle telematics (Wahlstrom, Skog and Handel, 2017)

Here we summarise some of the studies on the use of smartphones and IVDRs. Smartphones are widely used to collect data in the form of video, audio and IMU (acceleration, gyro, magnetometer, location). These data were used to detect driving behaviour and different road conditions as well. Smartphones are used in the majority of the studies in this review. According to Vlahogianni *et al*. (2017) and Walhstrom *et al.* (2017), smartphones offer a cheap, scalable, and easily implementable alternative to current road monitoring methods, although several methodological challenges remain. Engelbrecht *et al.* (2015) stated that the vehicle and navigation industry had gained new ways to collect data using the ever-growing worldwide penetration of smartphones. Several smartphone applications have been developed to detect driving behaviour and road anomalies (Dai *et al.*, 2010; Johnson and Trivedi, 2011; Castignani, Frank and Engel, 2013; Meseguer *et al.*, 2013; Stoichkov, 2013; Kalra, Chugh and Bansal, 2014; Pargaonkar *et al.*, 2014; D. Chen *et al.*, 2015; Khedkar and Ravi, 2015; Zabihi *et al.*, 2017). Smartphones are used as data collection, connectivity, and presentation devices (Meseguer *et al.*, 2013; Amarasinghe *et al.*, 2015).In contrast, Wahlstrom *et al.* (2014) discussed the difficulties faced in previous studies using smartphones to profile drivers and providing scores. Furthermore, they emphasised those studies using a smartphone as one of the sensory devices, data connection device and a feedback interface. They found that the limited data sampling rate affects the detection of some critical activities which happen suddenly.

IVDRs were used to detect various driving risk indices and provide feedback and Toledo, Musicant and Lotan (2008) found a significant difference in the rate of involvement in car crashes after installing the device. Even though IVDRs are more accurate than smartphones, it is still unaffordable due to its cost (RoSPA, 2013).

2.1.5.2 What are the sensors and features used to detect the behaviour?

The various sensors are used in driving behaviour and road anomaly detection. Accelerometer, Global Navigation Satellite Systems (GNSS), Gyroscope, Engine Control Unit (ECU) information from OBD II module (such as revolution per minute (RPM), Throttle position, Camera, and Microphone are some among those. References (Vaiana *et al.*, 2014), (Wahlström, Skog and Händel, 2014) used only GNSS location data to calculate the speed and acceleration using the backward difference method. Further, they stated that this method provides more accuracy in finding speed and acceleration. A framework was proposed to detect dangerous cornering events based on GNSS and IMU data in (Wahlström, Skog and Händel, 2015). The study conducted by Amarasinghe *et al.* (2015) only used OBD-II sensor data to detect reckless driving behaviour and vehicle anomalies. The speed, acceleration, and engine RPM were used as parameters by Meseguer *et al.* (2013). The total accuracy of 99.5% and 99.3% were achieved for the smartphone and OBD-II devices respectively under controlled experiments (Vlahogianni and Barmpounakis, 2017). D. Chen *et al.* (2015) developed a turning and cornering detection system on smartphones, where they used the gyroscope, accelerometer, GNSS, and Microphone sensor data. The microphone is used to detect the signal relay sound. Takeda et al. (2012) used numerous sensors to collect rich multimodal data which included 12-channel audio, four-channel video, GNSS information, gas and brake pedal pressure, steering angle, following distance, vehicle velocity, driver's heart rate, skin conductance, and emotion-based sweating on the palms and soles, etc. A recent study by Ferreira *et al.* (2017) on an investigation with different smartphone sensors and ML found that the accelerometer and gyroscope are the best sensors to detect driving behaviour. Sensor fusion is found to be more promising in driver behaviour and road anomaly detection (Wahlström, Skog and Händel, 2014; Castignani *et al.*, 2015b; Ferreira *et al.*, 2017).

2.1.5.3 What are the techniques used to detect or identify driving behaviour?

Table 2.5 presents the number of studies considered in their review. The studies were classified according to the sensors, features, and algorithms used to find the driving behaviour and/or road anomalies. This study gives us the variables we can use to detect driving behaviour and road anomalies.

Table 2.5: Summary of Variables and Algorithms used in a review on driving behaviour and/or road condition analysis using ML techniques (Wahlstrom, Skog and Handel, 2017)

		Number of Experiments	% in total reviews
Sensors	GNSS	26	50
	Accelerometer	41	79
	Gyroscope	21	40
	Magnetic	16	31
	OBD	7	13
	Integral Measurement Unit	5	9.6
Feature	High Acceleration	45	87
	High Cornering	43	83
	Low Cornering	21	40
Detection /Classification Algorithm	Threshold	35	67
	Support Vector Machine (SVM)	4	7.7
	Dynamic Time Wrapping (DTW)	7	13
	Hidden Markov Model (HMM)	1	1.9
	K-nearest neighbour	1	1.9
	K-means	2	3.8
	Naive Bayes classifier	3	5.8
	Symbolic Aggregation	1	1.9
	Pattern Matching	2	3.8

The majority of the research used a fixed or variable threshold-based driving detection, for example Lui et al. (2016). In Fazeen *et al.* (2012) threshold-based multiple-axis classification proved a better accuracy (85.6%) in detecting the road condition. Fuzzy logic based (Castignani *et al.*, 2015a) and SVM (Chen *et al.*, 2015; Hosseinioun, Al-Osman and Saddik, 2016; Wu, Zhang and Dong, 2016) proved some positive results among different ML Algorithms (MLAs).

A study (Saiprasert, Pholprasit and Thajchayapong, 2017) in 2017 found that pattern matching algorithm outperforms rule-based algorithms in detecting aggressive driving. A trail on comparing Dynamic Time Wrapping (DTW) and Maximum likelihood algorithms using a smartphone's accelerometer and gyroscope sensors revealed that Maximum-likelihood algorithm outperformed with 89.1% accuracy in determining aggressive driving where DTW performed with 84.5% accuracy (Engelbrecht *et al.*, 2015).

Chen *et al.* (2015) discussed and developed a real-time abnormal driving behaviour It was mentioned that 100% accuracy was obtained on Left/Right turn detection. Lane Change and Curvy road detection accuracy was 93% and 97% respectively when the phone was placed on the dashboard using the SVM classification algorithm. Ouyang et al. (2016) proposed a novel approach called MultiWave to detect cornering events on smartphones. It was mentioned that HMM-based or DTW-based systems are CPU and memory full solutions on the mobile platform. Bayesian network (BN) outperformed five other different MLAs such as k-nearest neighbour (k-NN), radial basis function network (RBFN), logistic model trees (LMT), multilayer perceptron (MP), and support vector machine (SVM). With Mesegeur et al. (2013), Neural Networks classified road conditions and driving styles with 98% and 77% accuracy respectively.

According to Hong, Margines and Dey (2014), a naïve Bayes classifier with 5-bin discretisation ML technique proved its aggressive driving style classification with an accuracy between 81% to 90.5%. With Ferreira *et al.* (2017), a quantitative evaluation of the performance of four different ML algorithms showed that Random Forest (RF) is by far the best performing MLA, followed by Multi-Layer Perceptron (MLP) than Support Vector Machine (SVM) and Bayesian network (BN); varying from 0.980 to 0.999 mean Area Under Curve (AUC) values.

2.1.5.4 What are the available driver feedback techniques?

Feedback is given in the form of Textural (In-dash messages, SMS, e-mails, web-based results) (Toledo, Musicant and Lotan, 2008; Farmer, Kirley and McCartt, 2010; Takeda *et* *al.*, 2011, 2012; Braun, Reiter and Siddharthan, 2015), Visual (Heads Up Display (HUD), warning lights and signs) (Tijerina *et al.*, 2000; Donmez, Boyle and Lee, 2007; Adell *et al.*, 2008; Simons-Morton *et al.*, 2013), Auditory (In-vehicle warning sounds (non-speech tones), and voice messages) (Cummings *et al.*, 2007; Kruger, Hefer and Matthew, 2013; Yan, Wang and Wu, 2016), and Haptic (Seat-mounted, steering wheel, pulsating pedals) (Tijerina *et al.*, 2000; Adell *et al.*, 2008; Asif *et al.*, 2009) forms.

The voice messages were rated significantly better than all the non-speech tones (Adell *et al.*, 2008). According to Belz, Robinson and Casali (1999), a multimodal solution, a combination of visual, auditory or haptic modes would probably be most efficient.

2.1.5.5 How does feedback affect driving behaviour?

In literature, the feedback on driving behaviour has shown a significant effect on driving. For example, Dogon et al. (2011) concluded that providing drivers with detailed information on what caused their failure or success and what they should do in order to improve their performance is needed for feedback to be effective in reducing self-enhancement biases. Feedback with positive content will encourage good driving behaviour (Braun, Reiter and Siddharthan, 2015),(Huang *et al.*, 2005).

It was stated that "Objective feedback about an elevated risk may encourage the driver to correct his or her driving"(Hong, Margines and Dey, 2014, p. 4054). In general, driving behaviour change in novice drivers was higher than veteran drivers using feedback from the technology. The technology affects driving safety both positively and negatively due to the distraction by the feedback systems (Lee, 2007) and sharing the driver's behaviour and performance with supporters positively affected driving behaviour (Zabihi *et al.*, 2017). Providing feedback to young drivers and their parents may reduce risky driving behaviour, but the success rate depends on the involvement of parents (Shimshoni *et al.*, 2015). "Simply installing the device in a teen's vehicle may not be sufficient to improve driving safety. However, providing video clips of safety-relevant driving behaviours to the teens and parents/guardian for review could create an opportunity for teens to learn from their mistakes" (Mcgehee *et al.*, 2007: 216). It was found that providing feedback to the parents with possible consequences reduced risky driving in teens, whereas no significant changes in driving behaviour were observed by providing immediate feedback only to teenagers (Simons-Morton *et al.*, 2013). Rewards are mostly considered in studies based on UBIs. It is evident that the insurance companies cannot afford to deduce the premium below a specific value, though it is impossible to restrict the drivers to stick to the rigid rules all the time as it will reduce the involvement in participating in the UBI programme.

Moreover, not all people like UBI because of their privacy concerns. Personalised feedback and more realistic positive feedbacks were preferred in (Huang *et al.*, 2005). Zabihi et al. (2017) found that not only social reinforcement affects drivers, but also their social face contributes in keeping good driving behaviour as drivers keep looking good to their closest person.

2.1.6 Discussion

There are few studies concerned with implementing algorithms to run on smartphones to detect activities in real-time. Such applications used threshold-based classification to detect aggressive and regular driving activity. Some studies proposed the adoptive profiling algorithms to set the threshold value to depend on various parameters. More studies on the Haptic feedback system was done using simulation-based trails, though only some of such studies are discussed here since most of those were simulation-based.

Only a few researchers like Zhao *et al.* (2013) have used ISO 2631-1-1997 standard to measure driving comfort in harsh driving conditions, while the majority of the researchers used their measurement standards using the feedback from the passengers. Position of the phone and its orientation was found to influence the accuracy of the detection. Coordinate transformation of the smartphone was done as a pre-processing before applying driving detection/classification algorithms. Only some researchers, notably Zheng, Member and Hansen (2016) emphasised that the effectiveness of coordinate transformation still needs to be studied further. Wide ranges of in-vehicle driver assistance system are available in the present automobile market, which is available as a retrofit or OEM products, but smartphones are still competing IVDRs due to its cost and convenience.

Wahlstrom, Skog and Handel (2017:16) stated that “Currently, the literature on these topics is very scattered, with many articles detailing ideas that have already been published elsewhere”.

2.1.7 Threats to Validity

Most of the studies employed a very few numbers of drivers, and a limited number of vehicles, which could lead to a biased result. Further, it was mentioned that experimenting aggressive driving in real road driving situation is dangerous (Wu, Zhang and Dong, 2016) and illegal, so most of the aggressive driving experiments were conducted in a controlled environment. Naturalistic driving data was used in some studies to check the correctness of the detections made; it was time-consuming due to human intervention.

Detection accuracy using smartphones varies due to the noise coming from the use of mobile phones for texting and talking during driving, which is not considered in most of the experiments. Only a few studies, for example *‘Driver behavior profiling using smartphones: A low-cost platform for driver monitoring’* (Castignani *et al.*, 2015a) clearly show that the calibration phase is more important to be focused on projects using mobile phones.

2.1.8 Conclusion of the SLR

More studies on presenting effective feedback techniques are needed. An efficient way of detecting who is using the phone while driving is a challenging task to be further studied. Only a few studies on total road safety monitoring which covered driver, vehicle and road anomaly have been done so far. Since there are different types of features used among each study, performing a meta-analysis is a challenging task.

2.1.9 Future Directions

Telematic data are becoming richer and more accurate due to the technological improvements in sensors and connectivity. IoV is becoming popular as a result of this. Use of machine learning and telematics data could be used to explore more about the road, vehicle, and driver. Inferring the payload or weight of a vehicle or finding an overloaded vehicle using telematics and ML is an exciting area to focus on.

2.2 WIM Systems

In general, WIM systems are used to measure the gross vehicle weight (GMW) and other parameters of vehicles (Gajda, Burnos and Sroka, 2018).

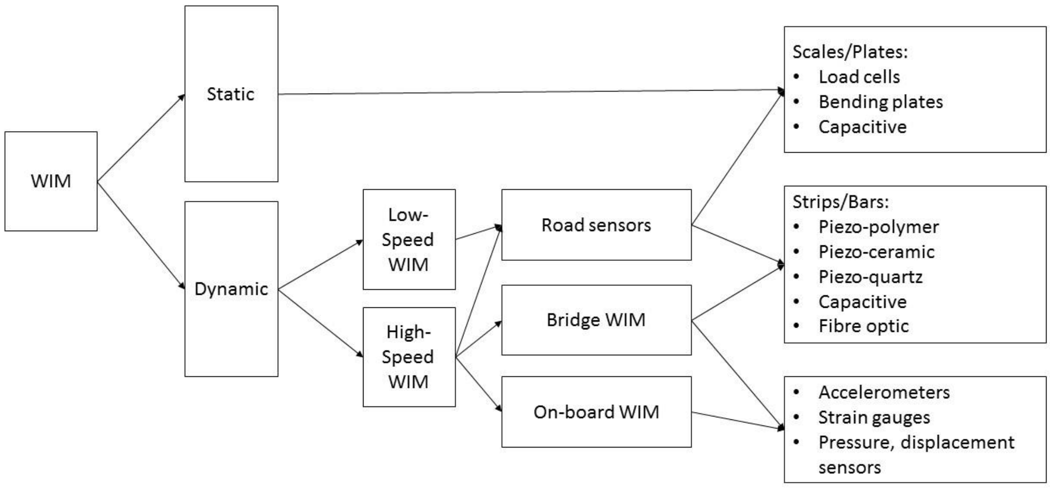


Figure 2.2: Classification of WIM (Labarrere, 2017)

Two main classifications of WIM solution methods are Static WIM and Dynamic WIM. There are two sub-categories of dynamic WIM methods, which are a) Low-speed WIM (LS-WIM) and b) HS-WIM (see Figure 2.2). In LS-WIM the vehicle is weighed while it moves across the scale at low speed, typically less than 15kmph, but HS-WIMs are capable of weighing the vehicle weight at full highway speeds (Richardson *et al.*, 2014).

2.2.1 Static WIM

In static WIM method, the vehicle is weighed while it is stationary on the scale. Static WIM methods are most accurate but cumbersome. Fixed Systems, Semi-Portable Systems, and Portable Systems are the three types of static WIMs in general (Richardson *et al.*, 2014). Fixed Systems are permanently mounted to the pavement, usually in a reinforced concrete frame or platform. Semi-Portable Systems use permanent grooves, and road installations with portable scales which are only installed while weighing operations are being carried out. Portable Systems use either wheel or axle scales, which are placed on the pavement surface (Richardson *et al.*, 2014).

2.2.2 Low-Speed WIM

According to Gajda, Burnos and Sroka (2018), Static scales and LS-WIM devices are very accurate and are used for enforcement in many US states and several European countries. LS-WIM devices were introduced because of the drawbacks of Static WIMs. LS-WIM devices are typically wheel or axle scales equipped with load cells and are usually installed into reinforced concrete or asphalt platforms which are at least 30-40m in length. The vehicle may be guided by curbs to minimise variation in the transverse position of the wheels. The data processing system analyses the signal from the load cells and takes the vehicle speed into account in order to accurately calculate wheel or axle loads.

LS-WIMs significantly reduced the time required to weigh vehicles, but it is not a feasible solution for highway deployments due to its cost for installation, maintenance and the significant delay in measuring the weight as the vehicles need to drive at slow speed. Static and LS-WIM systems require vehicles to exit the highway and wait in a queue. It is reported that this would delay between 10 and 30 minutes (Gajda, Burnos and Sroka, 2018).

2.2.3 High-Speed WIM

High-speed WIM systems are built to measure the weight of vehicles driving on highways. HS-WIMs calculate axle weights at full highway speed. Most of the HS-WIMs are unmanned. Therefore, it can collect data 24/7. These devices are either installed in the pavement or on the underside of a highway bridge. Several types of pavement-based HS-WIM devices exist, including bending plates, strip sensors, and multiple strip sensors. Alternatively, HS-WIM can be accomplished using bridge-weigh-in-motion (B-WIM) devices. Several factors influence the accuracy of B-WIM systems, thus the HS-WIM as well (Richardson *et al.*, 2014).

The Multi-Sensor WIM (MS-WIM) was introduced by Gajda, Burnos and Sroka (2018), because of the low accuracy or correctness of HS-WIMs. Gajda, Burnos and Sokra (2018) conclude that the measurement accuracy could be increased by incorporating MS-WIM in HS-WIM. They also pointed out that the cost of installing MS-WIM is a significant concern.

In a report by Al-Qadi *et al.* (2016), researchers have identified three significant factors that affect the accuracy of the WIM system, namely site condition, weather condition and vehicle characteristics. They reported that temperature and humidity could affect the accuracy of the sensors, which will impact the overall efficiency of the system. Among the other factors, site conditions, and pavement roughness have the most significant effect on the efficiency of the WIM system (Al-Qadi *et al.*, 2016). They also reported that vehicle characteristics, such as speed, tire type, inflation pressure, suspension system, and axle configurations, affect the dynamic tire force, thus affecting WIM sensor accuracy. According to Gajda *et al.* (2018), the HS-WIMs are still not as accurate as Static WIMs. Currently, the HS-WIMs are used to filter the overloaded vehicles from the traffic with limited certainty. The filtered vehicles are then sent to the Static WIMs for further legislative actions as it needs more accurate results (Gajda, Burnos and Sroka, 2018).

There are different practices in utilising WIM for enforcement over the world. Especially in Taiwan, HS-WIM solutions are utilised for direct enforcement, with tolerance up to 30% due to the errors in the framework. In some countries, portable HS-WIM systems are utilised over brief timeframes to recognise overloading, and afterwards, filtered vehicles were sent to perform weighing on static weighbridges. This is because the precision of portable WIM frameworks isn't generally excellent, and along these lines, the effectiveness or pre-determination is low (Jacob and Véronique, 2010).

2.2.4 Other Approaches

Several other approaches such as using the Tyre Pressure Monitoring Systems (TPMS), ride-height (suspension displacement), and chassis mounted scales have been proposed in addition to pavement based Static and Dynamic WIMs. Shah et al. (2016) developed a WIM system by observing the length of shock absorber in two-wheeler vehicles. A paper by Mckay *et al.* (2012) discusses an experiment to explore the various possibilities for passive WIM system. McKay *et al.* (2012) investigated multiple vehicle indicators including brake temperature, tire temperature, engine temperature, acceleration and deceleration rates, engine acoustics, suspension response, tire deformation and vibrational response. Their sensing system included; infrared video cameras, tri-axial accelerometers, microphones, video cameras and thermocouples. They found that the weight of a vehicle shows a strong correlation to tyre deflection, suspension response and some other features. The patent (Ihiguro *et al.*, 2006) discusses a vehicle weight estimation device. The invention is generally for estimating a vehicle’s weight used for determining a shift range of an automatic transmission vehicle. It is based on the acceleration integration and driving force integration. The torque value calculates the driving force, and the speed value is used to calculate the acceleration. Present vehicles use ECUs to compute the engine load and other values and adjust the parameters such as air intake, fuel injection, and ignition timing to increase the performance and efficiency (Paul Hilgeman and Vicente, 2000).

In summary, using mass scales in weighbridges is a costly and time-consuming solution. Chassis and seat-mounted scales use mechanical devices which need frequent calibration. Smart tyres and measuring weight using tyre pressure is expensive since it needs to be installed on every wheel. New WIM applications as a part of ITS solutions for traffic and vehicle load enforcement are expected. It would be easier to stop overloaded driving conditions by monitoring tucks using a reliable on-board WIM system. This can be achieved with efficient means of communications. Having such a system could save the installation and maintenance cost for traditional WIM systems (Kirushnath and Kabaso, 2018).

2.3 VT data collection module

This section discusses the background to the VT module by introducing some of the existing modules, and by describing the essential components of it. Several VT data collection modules are available in the market. Tselentis *et al.* (2017) summarised some of the VT data collection devices. Table 2.6 shows some of such devices available in 2017; adopted from Tselentis, Yannis and Vlahogianni (2017).

Table 2.6: Some of the Telematic devices and their cost (Tselentis et al., 2017)

Manufacturer	Data recorded	Method of transmission	Installation cost	Monthly /yearly fee
CarChipFleetPro	Distance, time, acceleration, speed, GPS location, fuel, Engine speed	USB cable/port (customer loaded)	$149 (plus a $395 charge for software, one per fleet) Can also be used wirelessly with a $200 base unit	None
Sky-meter	Time, distance, place, speed, acceleration of all driving, and the location and time of all parking	GPRS/CDMA (other protocols available at extra charge)	$50–$250 activation fee	$18.95 per month after one year
OnStar	Distance, speed, time, (incl. other features)	Automatic through GPS	First-year free for new GM cars (only available for GM)	$18.95 per month after one year
Freematics	Speed, distance, time, location, acceleration, engine RPM	Built-in Bluetooth Low Energy and SPP module for wireless data communication or via microSD card (32GB)	99$ (Plus $30 for GPS module, plus $10 for MEMS MPU-9150 (9-axis) module, plus $10 for DUO BLE-BT 2.1 and plus 5$ for 32GB microSD)	None
Progressive (MyRate Device)	Distance, speed, time, location, acceleration, trip frequency	Wirelessly	None but $75 fee if not timely returned at the end of the policy	Varies

2.3.1 Vehicle Electronic/Engine Control Unit (ECU)

Modern vehicles are equipped with several sensors and ECUs. The main reasons for these sensors and ECUs are to obtain performance with fuel efficiency and increased safety.

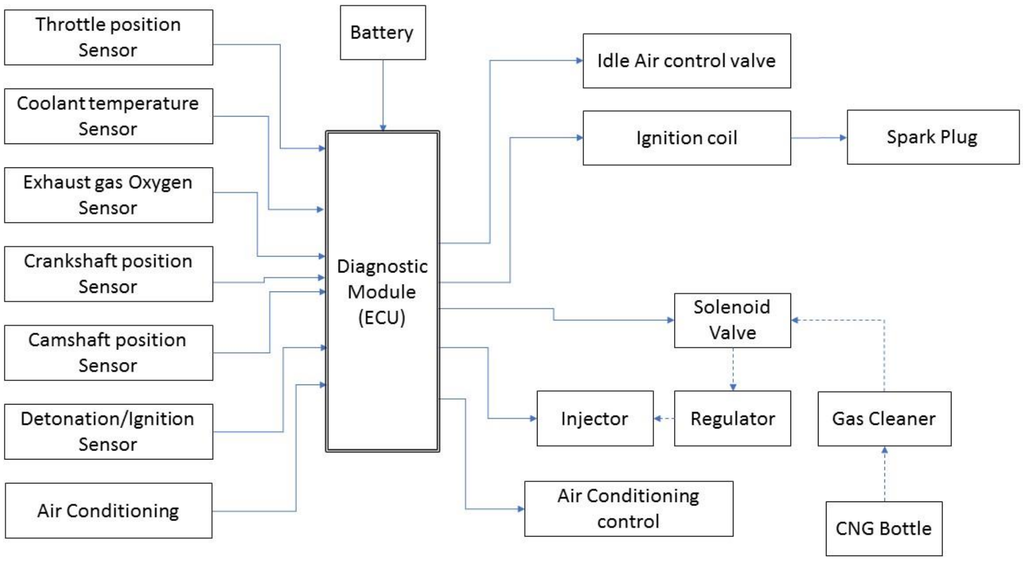


Figure 2.3: Schematic diagram of ECU with Sensors

The ECU receives the values from the array of sensors and interprets the values with a multi-dimensional performance map and controls the actuators accordingly (Massaro et al., 2016). Adjusting the air-fuel mixture and ignition timing for better combustion is one of the primary functions of the ECU. Controlling the Antilock Brake Sensor (ABS), and Air Bag are some of the safety functions with regards to safety. Numerous sensors are being used in autonomous vehicles. Figure 2.3 shows the schematic diagram of sensors and actuators connected to the ECU.

Table 2.7 contains some of the Control Systems, sensors and actuators in automobile vehicles.

Table 2.7: Example of some automotive control systems (Massaro et al., 2016)

Control System	Indirectly Controlled Variable	Directly Controlled Variable	Manipulated Variable	Sensor	Actuator
Fuel Injection System	Air-fuel ratio	Exhaust Oxygen Content	Quality of injection fuel	Zirconia and Titania based-electro chemical	Fuel Injector
Knock control	Knock	Knock sensor output	Ignition timing	Piezo-electric accelerometer	Ignition coil switch transistor
Anti-lock braking system	Wheel Slip limit	Wheel speed	Brake time pressure	Magnetic reluctance	ABS solenoid valve

2.3.2 Controller Area Network (CAN) Bus Data

The sensor data transmission happens via CAN bus to the ECU. With a large number of components that exchange data through a technology invented in 1986 by Robert Bosch

(Kiencke, Dais and Litschel, 1986), a serial broadcast bus that allows near-real-time management of most sensors and electronic devices embedded in the car (Massaro *et al.*, 2016). The CAN bus transmits ECU data outside for troubleshooting and performance logging using several standards, for example J1939.

2.3.3 Onboard Diagnostics (OBD) module

Almost all of the newly produced automobiles are required, by law, to provide an interface for the connection of diagnostic test equipment (Elmelectronics, 2011). Cars manufactured after 1996 have OBD interface which enables users to read and write into cars ECU. The communication protocol and the data transfer rate may differ from manufactures. ELM 327 is the widely used microcontroller (chip) for the communication with the ECU. The ELM 327 acts a bridge between OBD ports and a standard RS232 serial interface. ELM microchip has its interface to send and receive data from external applications and the ECUs designed by various manufacturers.

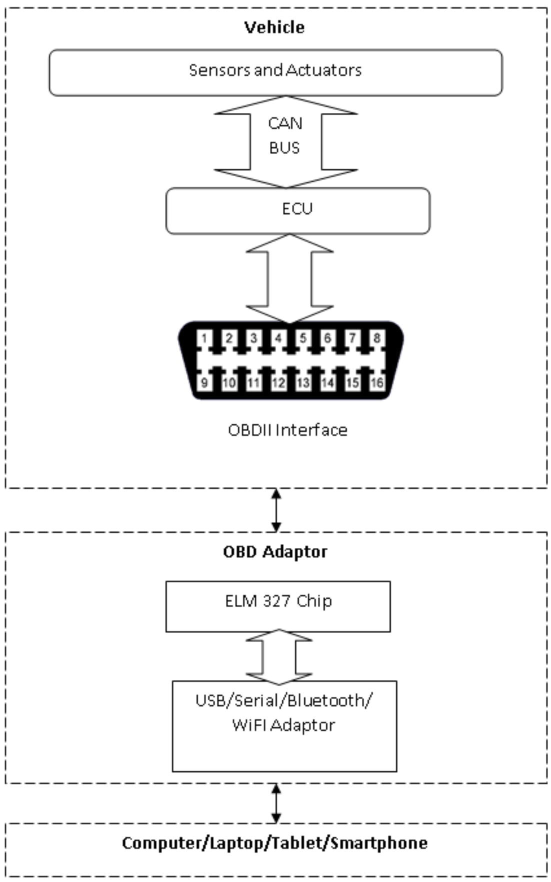


Figure 2.4: Block diagram showing ECU data collection modules

Figure 2.4 shows the block diagram of the modules involved in OBD data communication. The data from sensors to the ECU are transferred via CAN bus; vehicle manufacturers follow some standards on establishing communication to the external world through OBDII interface. They allow some generic parameters to be accessed by the OBD interface.

Once the OBD adaptor has been inserted to the vehicle's OBD interface, the ELM 327 tries to establish a connection with the vehicle's ECU. It tries with several communication protocols and baud rates to establish the connection. According to the ELM electronics report (Elmelectronics, 2011), there are 12 different protocols supported by ELM, including two user-defined protocols. After establishing the connection, the ELM 327 reads the data from the ECU and allows its connected applications to access the values by translating the ECU data. The communication can be made with several different modes (services). Table 2.8 describes the ten diagnostic services described in the latest OBD-II standard SAE J1979. Before 2002, J1979 referred to these services as "modes".

Table 2.8: The 10 diagnostic services/modes according to SAE J1979 standard

Service/Modes	Description
01	Show current data
02	Show freeze frame data
03	Show stored Diagnostic Trouble Codes
04	Clear Diagnostic Trouble Codes and stored values
05	Test results, oxygen sensor monitoring
06	Test results, other components/system monitoring
07	Show pending Diagnostic Trouble Codes
08	Control operation of on-board component/system
09	Request vehicle information
10	Permanent Diagnostic Trouble Codes (DTCs) (Cleared DTCs)

The parameters are accessed using their Parameter Identifiers (PIDs), for example, Engine Revolutions Per Minute (RPM) has the PID number 12 under service number 01. OBD-II was made mandatory for cars and lightweight trucks across the USA in 1996. OBD-II was required in the EU for all gasoline cars after 2001 followed by diesel in 2003. In 2005 it was required in the USA for medium trucks. In 2008 the ISO 15765-4 CAN bus standard was required in the USA. In 2010 OBD was required in the USA for all the Heavy-duty vehicles (CSS, 2018). In their article, Alessandrini *et al.* (2012) reported that OBD sensors have been validated and have good accuracies to be used to calculate instantaneous power and fuel consumption. This encouraged us to use the OBD 2 data in this project.

2.4 Theoretical Background

According to the Newtonians physics where space and time are absolute, we believe that Newton's theories of mechanics are still valid in this physical world. According to

Newton's second law of motion, "The acceleration produced when a force acts is directly proportional to the force and takes place in the direction in which the force acts", which is $F = ma$, in formula; where, F is the force applied on a mass, m, and a is the acceleration of the mass. This can also be interpreted that the applied force, F, is proportional to the mass, m, for a specific acceleration.

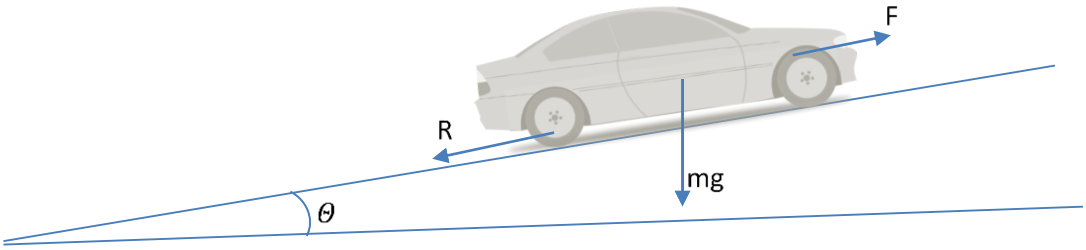


Figure 2.5: Forces on a moving vehicle

In vehicles powered with the internal combustion engines, the driving force, F, produced is proportional to the Torque (Engine Load) of the engine. Whereas, Torque is a function of Engine RPM and intake airflow. According to Mckay *et al.* (2012), the weight of a vehicle can be measured using several internal and external features. In Ihiguro *et al.* (2006), the estimating means of vehicle weight is based on the motion of the equation (see Figure 2.5).

$$m.dv = F - m \times g \times sin(\Theta) - R$$

Equation 2.1

Where, v = acceleration, m = vehicle mass, F = driving force, Θ = slope of the driving surface, g = gravitational acceleration, R = running resistance.

From Equation 2.1, the mass is:

$$m = \frac{F - R}{dv \times g \times sin(\Theta)}$$

Equation 2.2

Lin and Li (2016) listed the following as some of the conditions which affect a load of electric vehicles motor: 1) travelling surface, 2) road gradient, 3) weight of the vehicle, 4) rolling resistance, 5) type of tire, 6) air pressure of one or more tires, 7) air resistance, 8) size and shape of the vehicle, 9) alignment of wheels, 10) transmission system. Driving force of a vehicle affects the acceleration of a vehicle. The load is the amount of driving force needed to move a vehicle. In electric motors, the load is calculated using the ampere. According to SAE International SAE J1979 / ISO 15031-5, in internal combustion engines, the Calculated Engine load (EL) is a function of current airflow, ambient air temperature, RPM, peak airflow, and barometric pressure. According to SAE, EL is calculated using Equation 2.3, which is typically an indication of the current airflow divided by peak airflow at the wide-open throttle as a function of RPM, where airflow is corrected for altitude and ambient temperature (International Organization for Standardization, 2015).

$$EL = \frac{CAFR}{PAFR \times \quad (\frac{BARO}{29.92}) \times \sqrt{\frac{298}{AAT+273}}}$$

Equation 2.3

Where, CAFR = Current Air Flow Rate, PAFR = Peak Air Flow Rate at fully open throttle at standard temperature (25 °C) and pressure (29.92 in Hg BARO), AAT = Ambient Air Temperature (in °C).

In summary, Force, F, applied on an object with mass, M, produces an acceleration (Newtons' Second Law of Motion). This can also be viewed as the force needed to obtain a specific acceleration is proportional to the weight (mass) of an object. Internal combustion engine vehicles use the torque produced by engines to move the vehicle. The torque produced by the engine is proportional to the Calculated Engine Load, EL, given by Equation 2.3. Form these two factors; we could say that a vehicle's EL is proportional to the weight of it at a certain acceleration. But, EL is influenced by several internal (Equation 2.3) and external factors (Equation 2.2) (Ihiguro *et al.*, 2006; McKay *et al.*, 2012; Lin and Li, 2016). We assume that the relation is multiple linear regression. It can be viewed as:

$$\mathrm{W} = \mathrm{b} + \sum_{\mathrm{i}=0}^{\mathrm{n}} \mathrm{a_i} \times \mathrm{x_i}$$

Equation 2.4

Where, the weight of a vehicle W is the sum of a bias value b, and the accumulated sum of the products of coefficient ai and feature xi of all the n number of features.

2.5 Machine Learning

ML is a form of AI that enables a system to learn from data rather than through explicit programming. However, ML is not a simple process. ML uses a variety of algorithms that iteratively learn from data to improve, describe data, and predict outcomes. As the algorithms ingest training data, it is then possible to produce more precise models based on that data.

In general, the prediction is the primary goal of ML. Suppose T is a training set of N data:

$$T = (y_n, x_n), n = \{1, .., N\}$$

Equation 2.5

Where, y_n are the response (dependent) vectors and the x_n are vectors of predictor (independent) variables.

The goal is finding a function, f, operating on the space of prediction vectors with values in the response space, such that:

If the (y_n, x_n) are independent and identically distributed variable vectors from the distribution (Y, X) and given a loss function $L(y, \hat{y})$ that measures the loss between y and the prediction $\hat{y}$. The prediction error of using function f on training data T:

$$PE(f, T) = E_{Y,X}\, L(Y, f(X, T))$$

In the training process, we always try to choose f yielding small $PE(f, T)$ for a given data set T.

Typically, y is one dimensional. If y is numerical, the problem is referred to as regression (discussed below). If y is unordered labels or categorical values, the problem is called classification. The loss function in regression is usually squared error (discussed below). In classification, the loss is determined in binary values. The loss is one if the predicted category or label is not equal to the true (given) label, zero otherwise.

2.5.1 ML Model

An ML model is the output generated when the ML algorithm has trained with data. After training, when a model has provided with an input, an output will be given (Hurwitz and Kirsch, 2018).

2.5.2 ML algorithms

ML algorithms are organised into taxonomy, based on the desired outcome of the algorithm. Two primary classifications of ML algorithm types were Supervised Learning and Unsupervised Learning.

- **Supervised learning**

Supervised learning is where the algorithm generates a function that maps inputs to desired outputs. One standard formulation of the supervised learning task is the classification problem: the learner is required to learn (to approximate the behaviour of) a function which maps a vector into one of several classes by looking at several input-output examples of the function.

- **Unsupervised learning**

Unsupervised learning models a set of inputs: labelled examples are not available. These kinds of algorithms were commonly used in classification, clustering, and anomaly detection problems. The algorithm will learn the trends and variations in the input dataset and predicts the output automatically.

2.5.3 Supervised learning

Supervised ML comprises two main processes: classification and regression. Classification is the process where incoming data is labelled based on past data samples and manually trains the algorithm to recognise certain types of objects and categorise them accordingly. The system must know how to differentiate types of information, perform an optical character, image, or binary recognition. Regression is the process of identifying patterns and calculating the predictions of continuous outcomes. The system must understand the numbers, their values, and grouping (for example, heights and widths).

2.5.4 Linear Regression

Linear Regression is a simple model which makes it easily interpretable. A linear regression model assumes that the response or dependent variable (y) is a linear combination of weights (β's) multiplied by a set of predictor or independent variables (x). The complete formula contains an error term to account for random sampling noise ε.

$$y = \beta_0 + \sum_{\mathrm{n}=1}^{N} (\beta_n x_n) + \varepsilon$$

Equation 2.6

Where, β_0 is the intercept term, β_n is the coefficient of each predictor variable x_n from the N number of variables. The goal of learning a linear model from training data is to find the coefficients, β, that best explain the data. In linear regression, the best explanation is taken to mean the coefficients, β, that minimize the residual sum of squares (RSS), also known as Sum of Squared Error (SSE). RSS/SSE is the total of the squared differences between the known values, y_n, and the predicted model outputs $\hat{y}_n$. The residual sum of squares is a function of the model parameters.

$$RSS(\beta) = \sum_{n=1}^{N} (y_n - \hat{y}_n)^2$$

Equation 2.7

The coefficients β, which make the smallest RSS/SSE value, is obtained from the maximum likelihood estimate of β. This way of fitting the model by minimizing the RSS is called Ordinary Least Squares (OLS) (Pohlmann and Leitner, 2003).

Let $Y = (y_1, \ldots, y_N)^T$ be the response vector and X be the $N \times (p+1)$ matrix of covariates. Then the mean of Y is $X\beta$, and the OLS solution is:

$$\hat{\beta} = (X^T X)^{-1} X^T Y$$

Equation 2.8

OLS fit methods work well for single independent variable and single dependent variable regressions. If the response variable is in a non-linear relation with more than one predictor variables, the relation is called multiple non-linear regression or in some cases, multiple polynomial regression. It is simply achieved by introducing new variables by applying non-linear functions such as log, sin, square root, to the existing predictor variables. Gradient Descent method is commonly used to find the best coefficients (β) in multiple regressions. Standard regression methods are not very robust to outliers and nonlinearities and are prone to overfitting when the feature space is high-dimensional or when there are little training data (Chan and Vasconcelos, 2012).

2.5.5 Bayes' Theorem

If X and Y are two mutually exclusive events, given $P(X), P(Y)$ are the probabilities of X, Y respectively, then the conditional probability of Y given X is true , $P(Y|X)$ is :

$$P(Y|X) = \frac{P(X|Y)\, P(Y)}{\sum_{n=1}^{N} P(X|Y_n)\, P(Y_n)}$$

Equation 2.9

Where, $P(Y_n \cap X)$ for each Y_n or $P(X|Y_n)$ for each Y_n is known.

2.5.6 Bayesian Regression

Naïve Bayes for regression was discussed in the technical Note by Frank et al. (2000). Predicting the numeric target value Y, given an Example E. Where E consists N number of numeric attributes $X1, X2, X3, \dots, XN$. If the probability density function P(Y|E) of the target value Y was known for all examples E, the prediction error could be minimised by choosing Y. But, P(Y|E) is generally unknown, and it must be estimated from the data. Naïve Bayes uses Bayes theorem/rule by assuming that the attributes in E are independent given the target value Y.

In the Bayesian viewpoint, the formulation of linear regression uses probability distributions rather than point estimates. The response/target, Y, is not estimated as a single value but is assumed to be drawn from a probability distribution. The model for Bayesian Linear Regression with the response sampled from a normal distribution is:

$$Y \sim Normal(\beta^T X, \sigma^2 I)$$

Equation 2.10

The output, Y, is generated from a normal (Gaussian) Distribution characterised by a mean and variance. The mean for linear regression is the transpose of the weight matrix, β, multiplied by the predictor matrix, X. The variance is the square of the standard deviation σ (multiplied by the Identity matrix because this is a multi-dimensional formulation of the model).

The aim of Bayesian Linear Regression is not to find the single "best" value of the model parameters, but rather to determine the posterior distribution for the model parameters.

Not only is the response generated from a probability distribution, but the model parameters are assumed to come from distribution as well. The posterior probability of the model parameters is conditional upon the training inputs and outputs:

$$P(\beta|Y,X) = \frac{P(Y|\beta,X) \times P(\beta|X)}{P(Y|X)}$$

Equation 2.11

Here, P(β|y, X) is the posterior probability distribution of the model parameters given the inputs and outputs. This is equal to the likelihood of the data, P(y|β, X), multiplied by the prior probability of the parameters and divided by a normalisation constant.

In contrast to OLS, we have a posterior distribution for the model parameters that is proportional to the likelihood of the data multiplied by the prior probability of the parameters. Here we can observe the two primary benefits of Bayesian Linear Regression.

Priors: If we have domain knowledge or a guess for what the model parameters should be, we can include them in our model, unlike in the frequentist approach which assumes everything there is to know about the parameters comes from the data. If we do not have any estimates ahead of time, we can use non-informative priors for the parameters such as a normal distribution.

Posterior: The result of performing Bayesian Linear Regression is a distribution of possible model parameters based on the data and the prior. This allows us to quantify our uncertainty about the model: if we have fewer data points, the posterior distribution will be more spread out.

2.5.7 Decision Trees

A decision tree builds regression or classification models in the form of a tree structure. It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches, each representing values for the attribute tested. Leaf node represents a decision on the numerical target. The first decision node in a tree which corresponds to the best predictor is called a root node. Decision trees can handle both categorical and numerical data.

The core algorithm for building decision trees is called ID3 by J. R. Quinlan which employs a top-down, greedy search through the space of possible branches with no backtracking. ID3 algorithm with Standard Deviation Reduction is used for regression problems (i.e. for continuous target variables). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population.

2.5.8 Artificial Neural Network

An Artificial Neural Network (ANN) is a network of a large set of massively connected simple processing nodes (neurons) (Specht, 1991). ANN contains a connected (directed graph) architecture of nodes. The nodes are arranged into several layers. Generally, an ANN is a combination of three different layers namely, input, output, and hidden (see Figure 2.6). The nodes in the input layer hold the input variables. The output layer contains the nodes which represent the output value. The number of elements in the output layer depends on the problem (regression, binary/ multi-class classification). The hidden layer(s) contains nodes which are referred as neurons (YAO, 1999). Each neuron in the hidden/output layer learns the weight ($W_{i,j}$) of the arcs from the neurons in previous layer. The weight of each arc is based on the non-linear transfer function (i.e. activation function) f_i, where $f_i \epsilon \{gaussian, sigmoid, heaviside, softmax, RELU\}$ (Bircanoglu and Arica, 2018). The weights are learned by finding the best propagation of nodes neuron values from input to the output layer. The value of a neuron is calculated by the following equation.

$$a_k^i = f_i(\mathrm{b_i} + \sum_{k=1}^{m} a_k^{i-1} \cdot W_{\mathrm{i-1,k}})$$

Equation 2.12

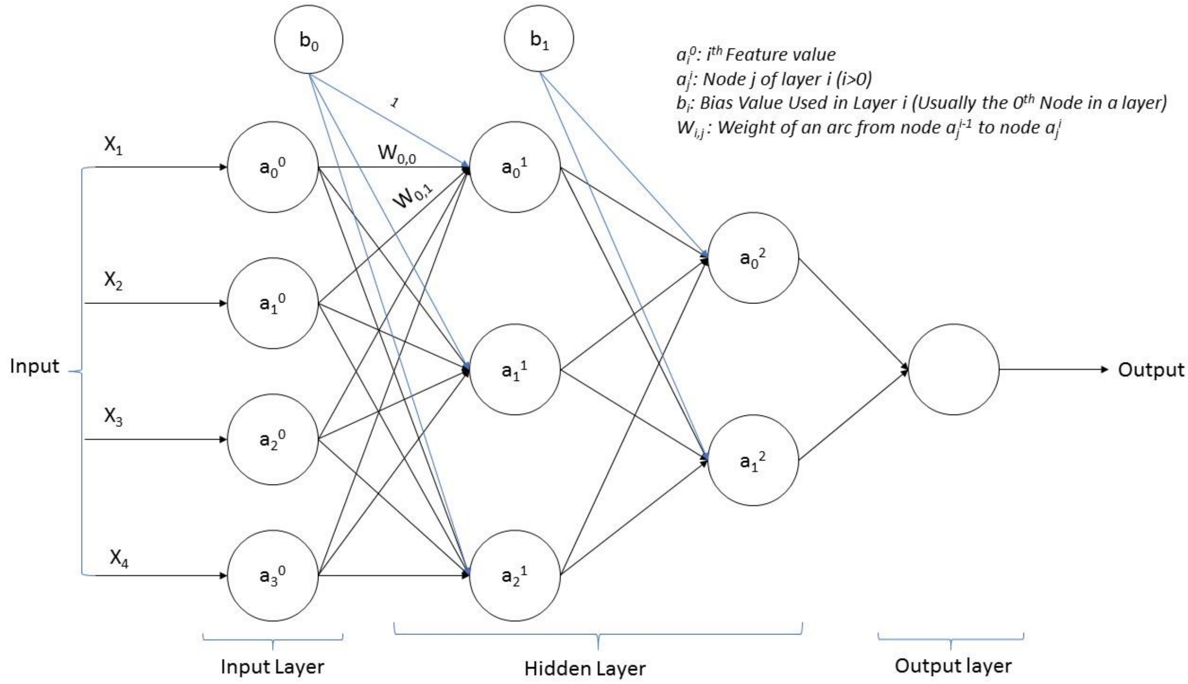


Figure 2.6: An ANN with two hidden layers

The ANNs with more than one hidden layer are called Deep Neural Networks (Akshay Kumar H and Suresh, 2016). There are two kinds of ANNs in use namely, Feedforward, and Recurrent. The Feedforward is the simplest form of neural networks which are also known as multilayer perceptron networks (MLP) (Razavi and Tolson, 2011). The goal of the MLPs is to approximate some function f_i. An MLP defines a mapping $y = f_i(X;\theta)$ and learns the value of θ (the matrix of weights $W_{i,j}$.) that results in the best function approximation. In Feedforward neural networks, the information flows through the function being evaluated from X, through the computations used to define f_i, and finally to the output y. There is no feedback connection from the output to feedback in Feedforward neural networks. Recurrent neural networks are the extension of Feedforward neural networks with the feedback connection. In this research, we use Feedforward neural networks.

2.6 Motivation

From the Systematic Literature Review discussed earlier in this chapter, it was evident that VT and ML are extensively used in road safety, mainly in identifying driving behaviour and road anomalies. And the use of VT data in vehicle condition monitoring still needs to be researched further. Meanwhile, several WIM systems are used to reduce/prevent overloading effects in many countries. Each WIM system has its own advantages and disadvantages. According to Newtonian's physics and literature, the gross weight of a motor vehicle is one of the factors affecting the required driving force

(Torque) of the vehicle. So, we could infer the gross weight of a vehicle if we know the torque and other factors. OBD interfaces are widely used to read engine data of vehicles. Smartphones and OBD devices have been used in many projects. ML algorithms offer us to infer value by training models. This dissertation discusses the effort taken to test the feasibility of using VT data and ML to infer the weight of a vehicle.

2.7 Summary

In this chapter, we have discussed the SLR on telematics and road safety. The SLR was done to find the current application of VT in road safety. After identifying the research area, we have discussed the VT data collection devices. Then we introduced the background knowledge related to the idea and motivation. The following chapter discusses the research philosophy and methodology.

CHAPTER 3

METHODOLOGY

This chapter discusses the philosophical viewpoint and the research methodology of this research. Answers to the following questions are discussed in this chapter.

Section 3.1. What is the type of research?

Section 3.2. What is the philosophical view of this research?

Section 3.3. How does this research fit into Design Science Research?

Section 3.4. What design theory is used in this research, and how is it applied?

Section 3.5. What were the reasoning techniques used in this research?

Section 3.6. How are the results of this research interpreted?

3.1 The Linear Model of Research

This research is applied research among the two research models: applied and basic. Where the basic research is more towards deriving a generalised solution than finding a solution for a real-world problem, the quest for fundamental understanding is high, but the consideration of use is low in basic research. The applied research tries solving problems in context by providing innovative solutions which are better than the existing solutions. The quest for understanding the fundamentals is low, and the consideration of use is high in applied research (Donald E. Stokes, 1997). Use-inspired basic research uses pure research findings and theories in practices to find an appropriate solution. Donald E. Stokes (1997) used Louis Pasteur's research as an example of the use-inspired basic research as in Figure 3.1.

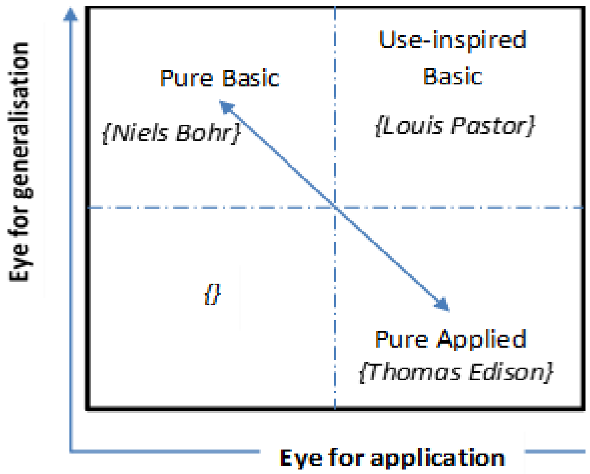


Figure 3.1: Pasteur's Quadrant and Liner model of research

In this research, we focused more on the application of technologies than on necessary background knowledge from pure research. Since this is research focused on the application in context than generalisation, we categorise this research under applied research than use-inspired basic research.

This research fits under the computer science domain. The application of computer science to solve some real-world problem was highlighted by Brooks who stated that., Computer Science is “not a science, but a synthetic, an engineering discipline. Computer Science is a type of engineering” (Camarinha-matos, 2012). In this research, we chose the tools and technologies to fit the requirements; for example, by keeping the current trend in mind, we used containerised application development. The selection of significant elements in this research was justified.

Schickore & Jutta (2014) argue that philosophy of science is exclusively concerned with the context of justification: “But like the AI-based theories of scientific discovery, methodologies of scientific discovery interpret the concept ‘discovery’ as a label for an extended process of generating and articulating new ideas and often describe the means concerning problem-solving. In these approaches, the distinction between the contexts of discovery and the context of justification is challenged because the methodology of discovery is understood to play a justificatory role”(Schickore and Jutta, 2014, chap. 8). Since the background of the research is enough for the context of discovery and as a science and engineering-based approach, this dissertation mainly focuses on the context of justification.

3.2 Philosophy

The primary objective of this research was to find the feasibility of using a new WIM solution through ML and VT. The truth we looked at is the gross weight of the vehicle (GWM) always remains the same regardless of the number of measurements we take at any place and time. As the definition of ontology is examining the nature of reality, in addressing the question of “what reality is?”, the correctness of the ML models was determined based on one single reality of truth from an ontological stance.

This research cascades under the *pragmatist* research philosophy, which deals with the facts and the choice of research philosophy, which is mostly determined by the research problem. In this research philosophy, practical results are considered important. In addition, pragmatism does not belong to any philosophical system and reality.

Researchers have the freedom to choose the methods, techniques, and procedures that best meet their needs and scientific research aims (Alghamdi, 2013). Pragmatists do not see the world as an absolute unity. The truth is what is currently in action. We used the tools which worked to fit the requirements.

3.2.1 The type of research from different viewpoints

Figure 3.2 illustrates the types of research from different perspectives, as specified by Ranjit (2011).

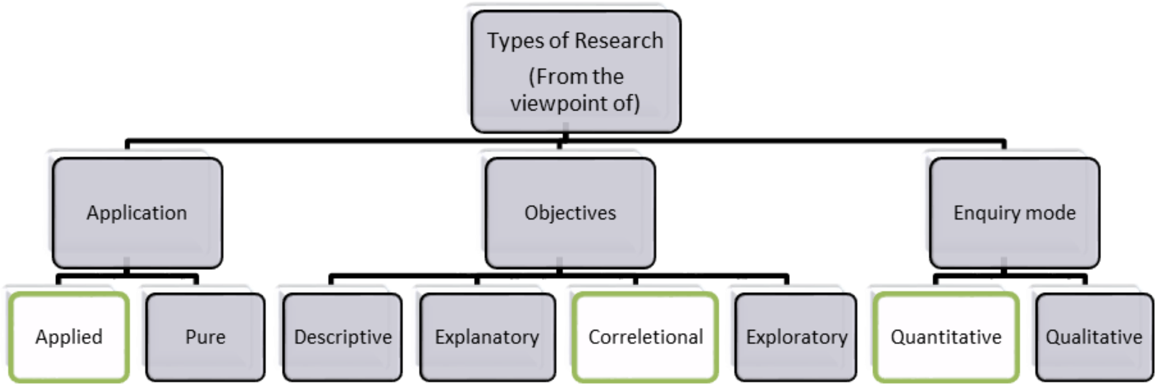


Figure 3.2: Types of research (Ranjit, 2011)

3.2.1.1 Viewpoint of application

This research falls under the view of Applied research type. Applied research is designed to solve practical problems of the real world, and often involves the use of some technology in the development of a new process or system (Camarinha-matos, 2012).

3.2.1.2 Viewpoint of Objectives

This research fits into the correlational research type. Correlational research discovers or establishes the existence of the relationship/association between two or more aspects of a situation (Camarinha-matos, 2012). The success of the expected system could assure the relationship between the weight of a vehicle with other internal and external parameters.

3.2.1.3 Viewpoint of Inquiry

This research is experimental research to determine the influence of different factors to infer the weight of a vehicle. Thus, this uses a quantitative strategy (or structured approach).

3.3 Research Methodology

The literature on research methodologies shows that research involved in solving a real-world problem with the design of an artefact is design research, also known as design science research. The Accreditation Board for Engineering and Technology's (ABET) definition states that engineering design is the process of devising a system, component, or process to meet desired needs. It is a decision-making process (often iterative), in which the basic sciences, mathematics, and engineering sciences are applied to optimally convert resources to meet a stated objective (Haik and Shahin, 2014). In this research, we designed and developed a new concept artefact and tested it under test conditions in a smaller laboratory scale.

3.4 Design Science Research

DSR is the design and investigation of artefact in context (Wieringa, 2014). The DSR changes the state-of-the-world through the finding of new artefacts. In this design science research, we design/develop an artefact with the aim to test the concept of a new WIM solution and to advance our knowledge about the characteristics of these artefacts and the processes to design and develop them.

During the DSR process, the problem statement is subject to change (Vaishnavi and Kuechler, 2004). However, "*the multiple world-states of the design science researcher are not the same as the multiple realities of the interpretive researcher: many if not most design science researchers believe in a single, stable underlying physical reality that constrains the multiplicity of world-states*" (Vaishnavi and Kuechler, 2004).

According to Wieringa (2016), there are two kinds of DSR projects, Problem-oriented, and Solution-oriented. *Problem-oriented* research, also known as evaluation research, learns about artefacts and how stakeholders use them by investigating the real world. Moreover, *solution-oriented* research is technical research which focuses on designing an artefact and validating it usually by simulations. This research is a *solution-oriented* design science research. The problem of this research is to design and validate the artefact. The implementation of the solution is the construction of a prototype in a test environment.

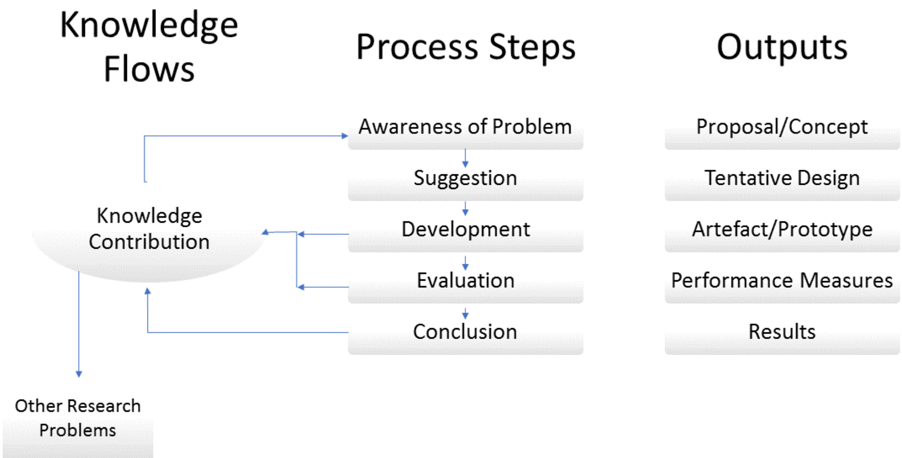


Figure 3.3: DSR Process Model (Vaishnavi and Kuechler, 2004)

Figure 3.3 shows the self-descriptive diagram explaining the process steps in DSR and the outputs of each step and the flow of knowledge by Vaishnavi and Kuechler (2004). They discussed five main DSR process steps. The awareness of the problem starts with the existing knowledge from the experience and the literature. New ideas and new concepts will become suggestions to solve the problems identified. The ideas will then be developed to produce an artefact. This development phase is an iterative process since the initial concept would be mostly abstract, and the specification may change during the design development process. The final artefact will then be tested in context. The conclusion would be the acceptability of the design suggestion based on its test performance. The DSR was used in various research. Mwilu *et al.* (2016), in their paper, presented a detailed topology of DSR artefact in Information Systems based on March & Smith (1995) and Hevner *et al.* (2004). Table 3.1 summarises the listed subcategories of the four main categories of DSR artefact namely, concepts of a construct, model, method, and instantiation.

Table 3.1: Categories of DSR artefact by Mwilu et al. (2016)

Constructs	Concepts	The concept is a new construct added to an extant language or meta-model
	Language	
	Meta-model	
Models	Ontologies	When constructs are used to build more structured objects, we obtain models.
	Taxonomies	
	Frameworks	
	Architecture	
	Requirements	
Methods	Guidelines	The method category puts together dynamic artefacts
	Algorithms	
	Method Fragment	
	Metric	
Instantiations	Implementations (Prototypes)	Instantiations are specific artefacts often proposed to assess the feasibility of other constructs
	Examples	

The artefact in this research is the Instantiation, i.e. a prototype of the proposed concept WIM system. A design research process is the application of scientific methods to the complex task of discovering solutions to the problems (Nunamaker, Chen and Purdin, 1990). Khandani (2005) listed five steps for solving design problems,1) Define the problem, 2) Gather pertinent information, 3) Generate multiple solutions, 4) Analyse and select a solution, 5) Test and implement the solution. Several other steps were also proposed in the literature. Offermann *et al.* (2009) subdivided design science design processes into three main phases namely, Problem Identification, Solution Design, and Evaluation. Table 3.2 discusses the several DSR processes discussed by different researchers and our research processes in each class.

Table 3.2: DSR Processes (Offermann et al., 2009)

	(Takeda, Veerkamp and Yoshikawa, 1990)	(Nunamaker, Chen and Purdin, 1990)	(March and Smith, 1995)	(Vaishnavi and Kuechler, 2004)	(Peffers *et al.*, 2008)	(Offermann *et al.*, 2009)	Our Approach
Problem Identification	• Enumeration of problems	• *Construct a Conceptual Framework (3)*		• Awareness of Problem	• Problem identification and motivation • Define the objectives for a solution	• *Identify the problem (2)* • *Literature research (1)* • Expert interviews • Pre-evaluate relevance	• 1,2,3
Solution design	• Suggestion Development	• *Develop a System Architecture* • Analyse & Design the System • Build the System	• *Build (2)*	• Suggestion • Development	• Design and development	• *Design artefact (1)* • Literature research	• 1,2
Evaluation	• Evaluation to confirm the solution • The decision on a solution to be adopted	• Observe & Evaluate the System	• Evaluate	• Evaluation • Conclusion	• *Demonstration (1)* • Evaluation	• Refine hypothesis • Expert survey • *Laboratory experiment(2)* • Case study / action research • Summarise result	• 1,2 in the context

Our DSR is a combination of processes specified by Nunamaker *et al.* (1990), March & Smith (1995), Peffers *et al.* (2008), and Offermann *et al.* (2009). Which is as follows:

A. Problem Identification
    1. Literature Research
    2. Identify Problem
    3. Construct a Conceptual Framework
B. Solution Design
    1. Design Artefact
    2. Build
C. Evaluation
    1. Evaluate
    2. Laboratory Experiment

In the problem identification phase, we have compiled a literature review (SLR) to identify the gaps in the use of ITS and identified the problem in transportation, and thereafter a conceptual framework was proposed. In the solution design phase, we designed the solution artefact as a prototype, and built it. Finally, in the evaluation phase, we evaluated and demonstrated the artefact in a context-specific context. The Problem identification phase is discussed in Chapters 1 and 2. The following Chapter (Chapter 4) discusses the solution design phase. The evaluation phase is discussed in Chapter 5.

3.4.1 Formation of design from knowledge

Unlike other research, design-oriented research such as DSR often begins with an abstract, vague idea (concept) from the researchers' mind. Structuring an analysis is reported to be much easier than formulating a well-structured definition of a design problem which may evolve through a series of steps or processes (phases) as we develop a complete understanding of the design problem (Khandani, 2005).

Takeda *et al.* (1990) used three factors which are prerequisites to describe a design process namely, 1) required specification (Ds), 2) design solutions (S), and 3) knowledge (K). They described the deductive design process logically as:

$$S \cup K \vdash Ds$$

Where, the design solutions, Ds, are derived from the specification, S, and the knowledge K. In the same paper, Takeda *et al.* (1990) discussed the abductive design process as:

$$Ds \cup K \vdash S$$

Where the design solutions and the knowledge about the design objects can be used to derive the design specifications, Abductive design process is considered an incremental process, in which the refinement of the design object takes place at each step of the abductive design process.

The deduction is used in obtaining the properties of the current solution with respect to the existing knowledge. Given the current design solution, Ds, design knowledge, Ko, and the required properties, P, as the specification are by:

$$Ds \cup Ko \vdash P$$

At each step of the design process, the deduction is applied to obtain all the properties of the current solution with respect to the currently available knowledge. This is to know the properties the current solution has and check whether the current solution satisfies the given specification and knowledge (Takeda, 1994).

The design research is incremental and flexible where the requirements and the views may change over time. Here at each step, the solution (concept) may change based on our experience (knowledge) on specifications within a context. This changing nature of the design process adds more knowledge. There must be a way of capturing these knowledge expansions in design science research. Several engineering research report in literature carried to convey the knowledge expansion process. There must be a way of capturing these knowledge expansions in design science research. A lot of research in literature carried to convey the knowledge expansion process. Notably, a unified design theory called Concept-Knowledge (C-K) theory was introduced by Hatchuel and Weli in 2003, which is then adopted in many engineering and science-based design research.

3.5 C-K Design Theory

In 1996 Hatchuel drafted, and in the early 2000s with Weli introduced, a unified design theory called Concept-Knowledge (C-K) theory. “The name ‘C-K theory’ mirrors the assumption that design can be modelled as the interplay between two interdependent spaces having different structures and logics: the space of concepts (C) and the space of knowledge (K). The structures of these two spaces determine the core propositions of C-K theory” (Hatchuel and Weil, 2007).

The Knowledge space (K) includes all established propositions which are true. The K space holds available knowledge such as scientific and engineering models and facts, physical laws. Concept space contains the vague concepts (ideas) which are un-decidable (neither true nor false) about some partially unknown set of objects x called C-set. Concept space C corresponds to the incomplete description of objects. The partial description of objects in C space captures the notion of design briefs or broad specifications. In essence, Concept space C holds two sets, pragmatic notion of brief or broad specifications we find in innovative design, unusual sets of objects x. Concepts are propositions of the form : “There exists some object x, for which a group of properties p1, p2, pk hold in K” (i.e. P(x) : Properties of $x \in K$, $\exists x\ P(x)$). All elements building the propositions in C come from K but do not belong to K (Hatchuel and Weil, 2007).

The design process can be described then by the interaction and co-evolution of these two spaces C and K through the application of four types of operators; $C \rightarrow K, K \rightarrow C, K \rightarrow K$ and $C \rightarrow C$ (described below). Design partitions the sub-concepts of C space by adding or deleting properties that arose from K space. There are two kinds of partitioning namely, Restrictive and Expansive. Restrictive partitioning adds usual properties of the object. Expansive partitioning adds new properties (sub-concepts). The partitioning of a concept may result in an expansion of K space. This could happen due to the learning of new knowledge to pursue creative expansion of C space, or the experience from one concept design phase (Hatchuel and Weil, 2002). The feasibility of the available objects in C space cannot be determined with the available knowledge. The design process tends to expand the C and K spaces simultaneously. On one side of the expansion, there are new creative concepts, whereas on the other side, there's new learning knowledge allowing the realisation of concepts. Design ends when the properties introduced into the concept can be validated in K space; that is it can be confirmed in K that such an object may exist (Hatchuel and Weil, 2007; Kazakci, Hatchuel and Weil, 2008). It is shown that C-K theory is sufficient to describe the generation of new objects and new knowledge which are distinctive features of design (Hatchuel and Weil, 2009).

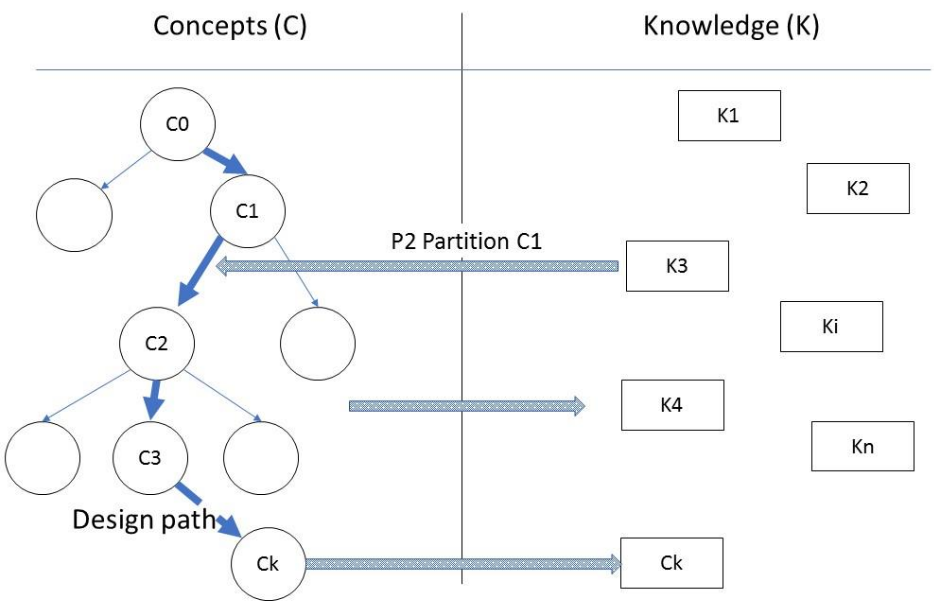


Figure 3.4: C-K Diagram adopted from Hatchuel and Weil (2007)

Figure 3.4 illustrates the C-K diagram presented in Hatchuel and Weil (2007).The concept C0 from existing knowledge K1 is used to test the new concept C1. A new knowledge K2 is added after C1, C2 will be formed and tested using K2. The design phase continues until the end of the building and testing the artefact.

3.5.1 The C-K Design Square

The design process always tries to find a better solution for the given specification in context. This helps the two spaces, C, and K, to expand during the course of the design. As mentioned earlier, in design, each space helps the other to expand. According to Hatchuel *et al.* (2004), the design process is nothing more than the operators that allow these two spaces to expand. And there are necessarily four different kinds of operators: the external ones: $C \to K,\ K \to C$; the internal ones: $C \to C,\ K \to K$.

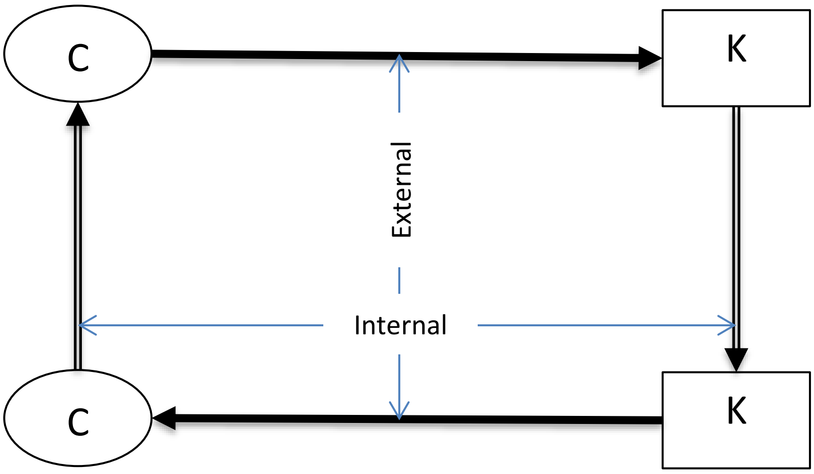


Figure 3.5: The C-K Design Square (Hatchuel, Masson and Weil, 2004)

Figure 3.5 shows the C-K design square of four operators (Hatchuel, Masson and Weil, 2004). In the design process, a concept may generate another concept, or it may transform into knowledge. The design process always seeks to expand the concept space (ΔC), with existing knowledge (Ko) through disjunctive ($K \rightarrow C$) operators. Also expands Knowledge space (ΔK) with existing concepts (C) using conjunctive operators ($C \rightarrow K$) (Hatchuel *et al.*, 2017)

External operators:

$K \to C$: Here, the properties will be added or subtracted from K to concepts in C. This operator creates *disjunctions* when it transforms a proposition into a concept. Which usually generate new alternatives; these alternatives are not concepts but potential seeds for alternatives. This operator expands space C with elements from space K (Armand Hatchuel and B Weil, 2003). This was performed at every stage where we came up with a possible set of solutions. For example, in this research, there were two possibilities to use for collecting data namely, custom-building the black-box device, and using a smartphone with OBD-II adaptor. Those possible alternative sub-solutions (Cn*(x)) were derived from the knowledge from the literature search.

$C \rightarrow K$: this operator seeks for properties in K that could be added or subtracted to reach propositions with a logical status. The validity of the alternative during $K \rightarrow C$ operation contributes to knowledge. It creates *conjunctions* which could be accepted as finished designs. The validation of a design concept by doing a test, prototype, a mock-up are examples of $C \rightarrow K$ operators. This operator expands knowledge with the help of concepts (Hatchuel and Weil, 2003). For example, in this research, we had several ML algorithms as alternatives to infer the vehicle weight. Neural Network, Bayesian, Decision Trees, and Linear Regression were some of the alternative algorithms used for the inference. Testing of each algorithm within our context leads us to select one best algorithm (ANN), which performed better than the rest of the possible set. This updated our understanding of the behaviour of several candidates' algorithms in context.

Internal operators:

$C \rightarrow C$: This operator is at least the classical rules in set theory that controls partition or inclusion (Hatchuel and Weil, 2003). As shown in Figure 3.4, a new partition or branch of concept (sub-concept, Cn*) will be added when we test a concept and accept it (Cn). The new branch will be possibly partitioned if necessary. In general, a design solution is an artefact. The artefact is a combination of several sub-modules. If we consider given examples for external operators above, the overall artefact is an eco-system of data collection devices and the ML backend. After choosing the smartphone with OBD-II adaptor as the data collection device (in $K \rightarrow C$), we tested it to verify by collecting some sample data. This confirmed the selection of the data collection device ($C \rightarrow K$). Next, we moved to a new branch (partition, $C \rightarrow C$) focusing on the selection of ML algorithms for the backend.

$K \rightarrow K$: this operator is at least the classical rules of logic and propositional calculus that allow a knowledge space to have a self- expansion. Proving new theorems, generalising and formulating new hypothesis are some of the activities from this operator. This would be based on the expansion of existing knowledge to new knowledge (experience) from the development of the design artefact.

Hatchuel & Weil (2007b) documented two major benefits of C-K theory in real research and development, better control of the design rationale, an increase of the innovative power of the design work. The second benefit usually implies the first one. Shifting the research direction during the research and development process is common in design research. For example, in our DSR project, the research approaches and directions were shifted at different stages during development. Hatchuel & Weil (2007) stated that such shifts appeared easily understandable with C-K theory because they were the joint consequence of both concept and knowledge expansions.

Hatchuel *et al.* in their recent paper in 2017 concluded that *"C-K theory appears today as a solid scientific ground for a transdisciplinary shift. Creative processes are better understood and modelled within Design theory and science. Then, such new science can contribute to research on human activities that were already seen as creative; it can also help to study creative forms in domains where they are less visible or hidden. Finally, creative thinking is no more reduced to psychological and natural phenomena, and it reveals a forgotten class of scientific thinking, the generic design of unknown objects and its co-expansion with the transformation of knowledge. Through the formalization of C-K theory, such a paradigmatic shift has already opened new ways of research and provided unexpected findings. Yet, all this could be only the early steps of a much wider scientific impact."* (Hatchuel *et al.*, 2017, p. 11).

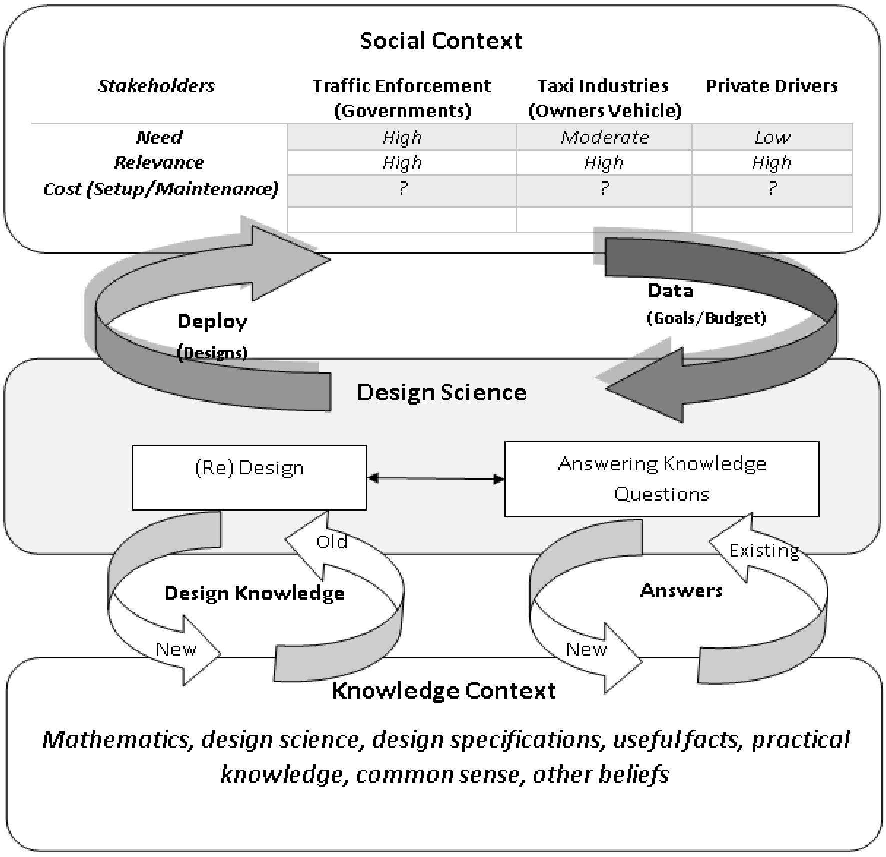


Figure 3.6: DSR for Social Context using Knowledge Context adopted from (Wieringa, 2016)

Figure 3.6 shows the relationship between the design science, knowledge and social context. According to Wieringa (2016), the DSR tries to solve a problem in society, by designing and deploying an artefact in the context. The design phase uses existing background knowledge, and knowledge gets updated over time. The C-K theory was adopted in this research as a framework to track and record the concept of design development. We also proposed and used a slightly different method for recording the design process.

Application of C-K theory in Science-based research was illustrated using the design and development of a new combustion system for Martian spacecraft. The paper (Hatchuel, Masson and Weil, 2004), explained the use of C-K theory in the prototype design of the mars rocket project by European Space Agency (ESA). The same example was discussed in other papers (Hatchuel *et al.*, 2004; Hatchuel, Masson and Weil, 2006; Hatchuel and Weil, 2009). An industrial application of C-K theory was discussed by Hatchuel, Masson and Weil (2006). The authors discussed the design of new bio-climate in cars. In that paper, they discussed three main factors, how C-K modelling accounts the explorations in a specific industrial situation, how C-K theory helps to understand the main design spaces, and how it enables to monitor the exploration process. Use of C-K theory in IT-based DSR is not visible in literature.

We can view the final C-K tree as a Depth First Spanning Tree, where the levels are of the design spaces, and the nodes are the concepts in each design space. The C-K theory was used as a skeleton in the design portion of our research. During the design, the concepts were designed with existing knowledge and the knowledge was updated incrementally.

3.6 Recording concept design

Here, we introduce a new method to represent design process, called *Concept Tree*. The notions we used are based on original C-K theory. We predominantly focus on the C space (concepts/ideas) in our C-K design recording method.

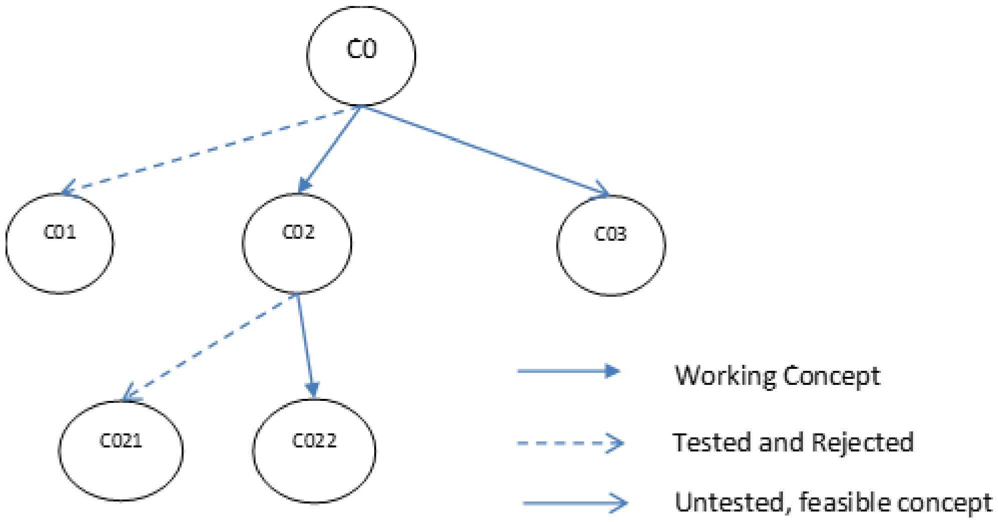


Figure 3.7: Recording design concepts (Concept Tree building)

The exemplary graph of the proposed design recording is given in Figure 3.7.

This method contains the following rules:

- **All sibling nodes were the possible design solutions at that level (design space):**

  Each new design space (level) starts from its predecessor. Thus, the design concepts are restricted to its predecessor.

- **The concepts must be tested from left to right:**

  Adaptation (tried/tested/implemented) concepts must be reported from left to right. This would allow designers to share their ideas which they tried first. They can move to the next possible idea if the previous one failed and choose the next best working concept.

- **Three types of arcs (arrows):**

  Depending on the implementation/testing, the concept, the possible alternative concepts may become unusable, useable. A concept is said unimplemented when we do not try to test or implement the concept. There are possibilities where we generate new ideas but only pick one (try) to implement (a common pragmatist approach) and forward to the next design space. The untested concept opens space for future research. The three different node types are addressed by different type of arrows in the graph (see Figure 3.7).

Two classes of Nodes:

The circles in the C-K graph represent concepts in general. But the initial concept, C0, is the root node of the tree which has propositions which are just vague statements. These propositions add the boundary to the thinking of the designer. This restricts the size of the possible design solutions at the next level. Each level is a new design space.

The nodes (not C0) in each design space is an alternative candidate solution idea based on its predecessor. This method was used to demonstrate the use of C-K theory in this research and was described in Chapter 5.

3.7 Reasoning

The phenomena studied in IT research are artefacts that are designed and built by a human to achieve the purposes of a human (March and Gerald, 1995). Implications for IT research are threefold. First, there may not be an underlying deep structure to support a theory of IT. Our theories may need, instead, to be based on theories of the natural phenomena (i.e., human) that are impacted by the technology. Second, our artefacts are perishable. Hence our research results are perishable. As needs change, the artefacts produced to meet those needs also change, which brings a theory of how programmers use a now-defunct language. Third, the IT artefacts are produced at an ever-increasing rate, resulting in numerous phenomena to study. Explicating and evaluating IT artefacts (constructs, models, methods, and instantiations) will facilitate their categorization so that research efforts will not be wasted building and studying artefacts that have already been built and studied "in kind." (March and Gerald, 1995).

The artefact was developed with the aim of testing our idea. The developed artefact was then used to draw our case-based conclusion. The development phase of the artefact comes under the design and engineering cycles of the design science research. This phase was carried out by the sequence of design cycles to refine the global design. Each design phase has its own reasoning type.

According to Lu and Liu (2012), a design process consists of three reasonings, inductive, deductive, and abductive. In the inductive type of reasoning, we come to a general conclusion from a specific observation. Similarly, in deductive reasoning, we make our conclusion from applicable rules. But, in abductive reasoning, we hypothesise based on some incomplete or smaller set of observations. Additionally, to minimise the logical extension and to avoid dealing with exceptions, the Circumscription (McCarthy, 1980) is used. “Circumscription is a type of commonsense reasoning and has been developed to deal with *exceptions.* In circumscription, exceptions for a given context can be determined by minimizing logical extensions of the predicates which represent *abnormality* with keeping the whole context consistent. Here abnormality is the implicit description of each piece of knowledge” (Takeda, 1994, p. 5).

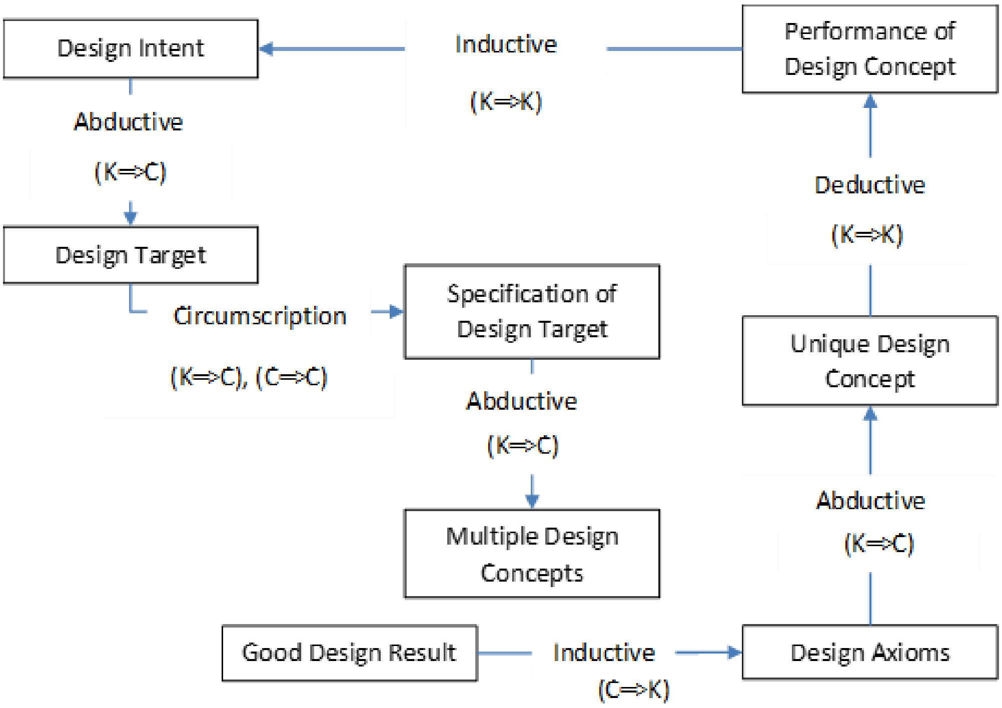


Figure 3.8: Reasoning in Design adapted from (Lu and Liu, 2012)

Figure 3.8 shows the diagram of the design research processes, and the inferencing used at each step with the C-K operation on each step.

The application was developed to infer the weight of different vehicles, but the research carried out in the context of a single vehicle to investigate a single case scenario, that is to see how the developed artefact performs on a single case scenario. The training and test data were gathered from one vehicle. The artefact was tested in the context of a vehicle. This phase of the research was carried out as deductive research, which also carried as a proof of concept. In general, inductive and deductive reasoning was used in this research. Moreover, the solution this research discusses is not a definite solution but a feasible one.

3.8 Facts to Generalization vs Facts to Explanation

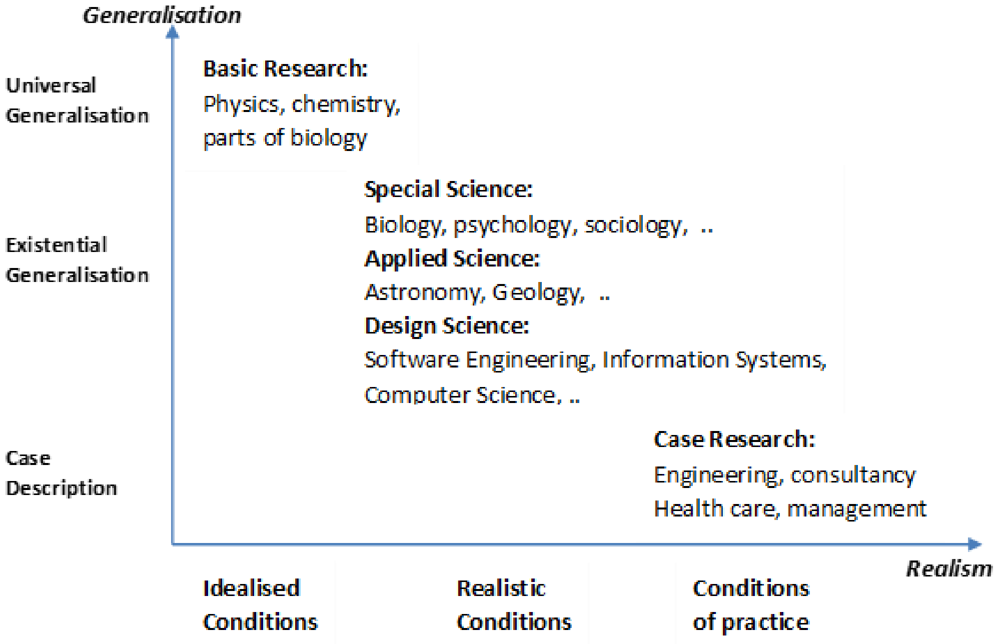


Figure 3.9: Position of research in the Realism vs Generalisation plane (Wieringa and Daneva, 2015)

Figure 3.9 shows a list of different kinds of research. As we discussed earlier in this chapter, the basic research is done in more idealised conditions, and they form general theories. In contrast, the case research is more domain-specific and more towards solving a specific problem. This research, a design science research, was done in realistic conditions. The prototype was developed by accepting the working concepts and moved forward as all other pragmatists do. The system was tested in the context of small, gasoline (petrol) cars. The findings of the model testing cannot be generalised to all motor vehicles. However, it can be said that if it works on a motor vehicle, and all motor vehicles share the common properties we looked at, then it would also work on other motor vehicles. It may not be true in some cases; further empirical evidence is required in the larger context.

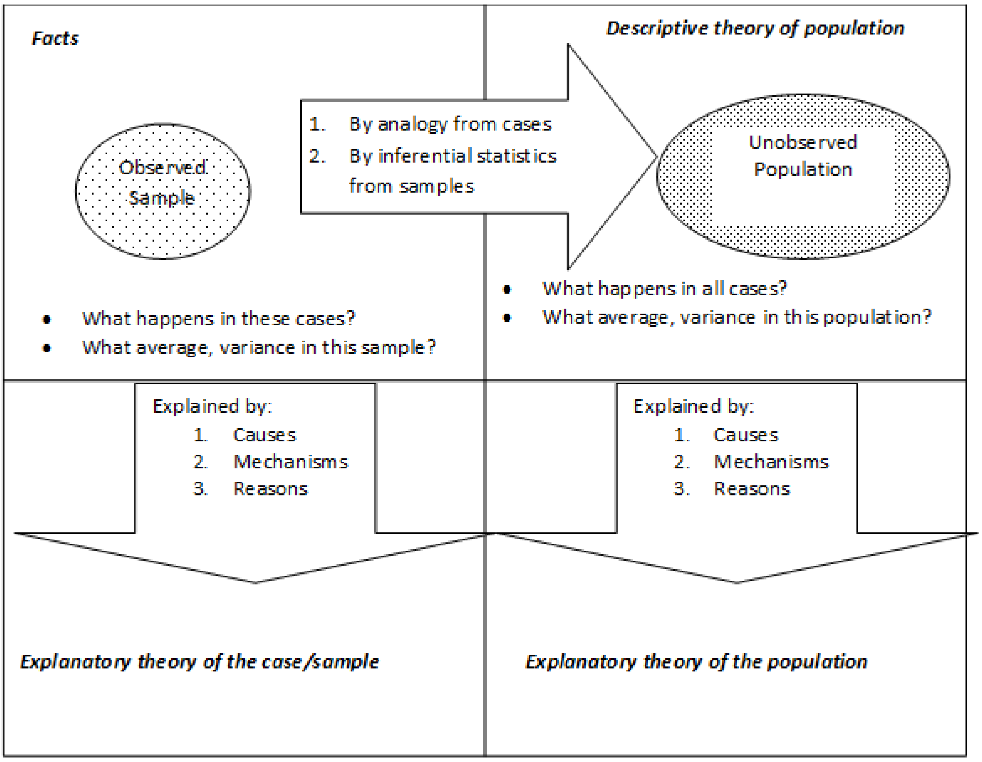


Figure 3.10: Two way to go beyond facts (Wieringa, 2016)

Figure 3.10 adapted from Wieringa (2016) shows the two pathways from facts. The developed artefact was tested on a small scale using a single motor vehicle's data. Thus, it cannot be generalised. This research tries to explain the design and development of the artefact and the performance of the artefact in terms of prediction accuracy. This research can then be repeated with many vehicles and be generalised in future.

3.9 Summary

In this chapter, we have discussed the common practices in DSR research. Considering the fact that majority of the articles produced from DSR were not focused on reporting the design of the prototype as a DSR artefact, the use and appropriateness of C-K design theory, in the context of a DSR producing prototype, was discussed. We propose a new DSR recording strategy named *Concept Tree* based on C-K theory. The example used in this research is the prototype design of a new WIM approach using VT and ML. Concept Tree built for the prototype WIM system design is discussed in Chapter 5. The following chapter introduces the design and development of the prototype system.

CHAPTER 4
SYSTEM DESIGN

This chapter discusses the overall system design starting from the illustration of the design of the conceptual framework to the selection of the ML models, and the evaluation criteria.

Section 4.1. The Conceptual Design Framework
Section 4.2. Design Considerations
Section 4.3. Prototype System
Section 4.4. Data Collection
Section 4.5. Correctness of Data
Section 4.6. Data Pre-processing
Section 4.7. Data Transmission
Section 4.8. ML Model Selection
Section 4.9. Evaluation

4.1 The Conceptual Design Framework of the WIM Application

According to the C-K design theory, our initial concept (C0) of this research is a new WIM system that is much easier (p1), faster (p2), and omnipresent (p3). Our knowledge (K0) to make a new idea from C0 was obtained from the fundamental laws of physics and some research discussed in Chapter 2. The new design idea using VT an ML is reported in this chapter. In this section of the design framework, we discuss the design requirements and consideration of the designed artefact, the WIM application.

An application (WIM Application) was developed as a by-product of this research. The primary purpose of the application was to read VT data from several sources (VT modules), and train and test the inference model from the data received.

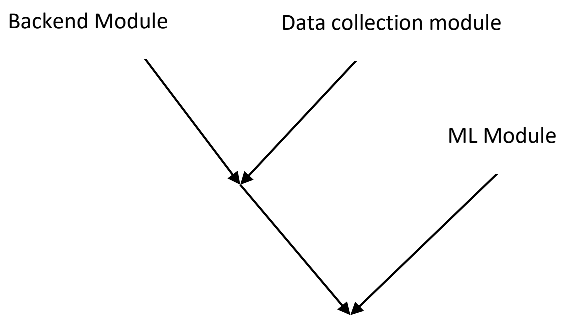


Figure 4.1: Dependency graph of the main design modules of the artefact

The conceptual design framework of the WIM application was proposed considering three modules namely, the back-end module, data collection module, and ML module. The first two modules were independent, and the third module was dependent on the other two modules. Figure 4.1 shows the order of the design and development performed, considering their dependencies.

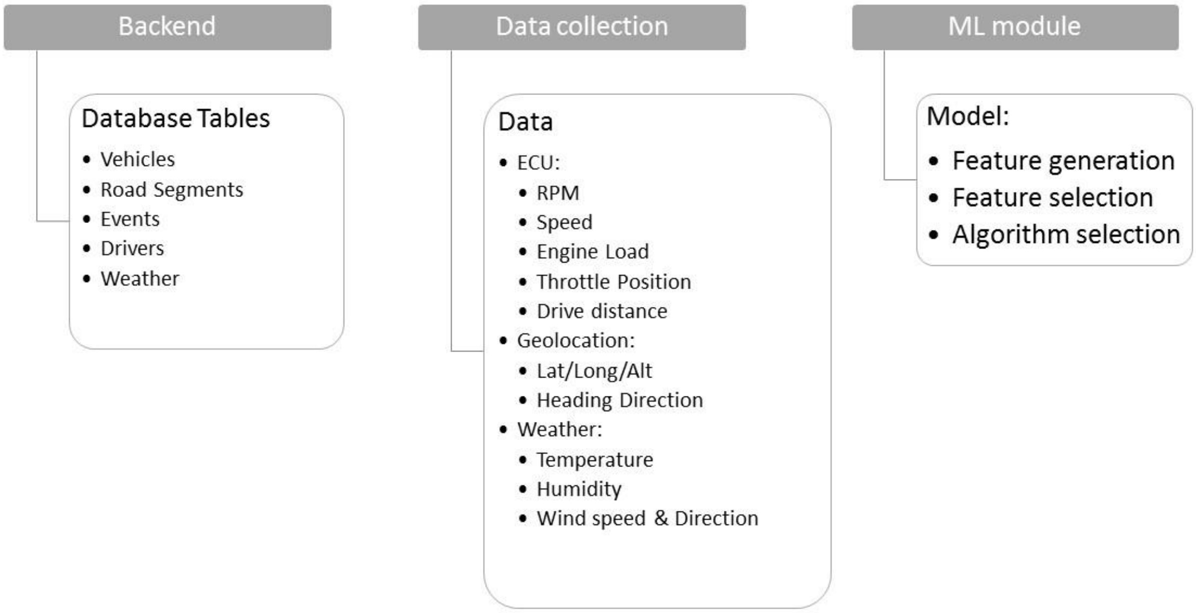


Figure 4.2: Overview of the proposed WIM system components

The backend was designed to receive weather data, simple and bulk (collection of) VT data from the data collection module and store it into a database. The database was also designed to log the inferred weight data from various driving events; this includes the year, month, time, vehicle identifier, start location, end location, and the inferred payload of each registered vehicle.

The data collection module was designed to receive VT data from OBDII devices, geolocation data from GNSS modules, IMU data, and weather data. It is also used to send the received data to the backend. New sub-concepts (or branches) from initial concept C0 arose in every stage of development, starting from choosing the deployment platform, better programming language, and others. It was decided to build a cloud-compatible (cloud-native) application; it became our new subset concept C0x in design space x.

4.2 Design Consideration

The artefact is divided into three major modules namely, data collection module, backend module, and ML module. The following sections of this chapter discuss the following:

Backend

- How to build a backend to read and store for the inference system?
- What architectures support our requirements?

Data collection Module

- What features to collect?
- How to collect all the features?
- How to store and transfer data?

Building and Training Model

- What feature engineering is to be performed?
- What are the candidate models for the problem?
- What are the performances of the chosen models?

4.2.1 Big Data from IoT devices

The VT data collection devices can be categorised under IoV devices, where the IoV devices are IoT devices which transmit vehicular data. The data collection module collects the GNSS and OBD data every second. There will be tons of such devices in a real-world scenario. Each IoV device will send the VT data either as a stream or batch to the cloud-native backend. Steam is a transmission of continuous data in real-time - this makes big data fast data. An enormous amount of data will be reaching the backend at the same time. This brings some design considerations in the means of communication, security and load handling.

Communication

Each VT data collection devices (IoV) produce telematics data at a rate of 1 Hz. If we assume that it sends the data to the backend as it receives, then this would become a stream data. Unlike batch data, stream data does not have a predetermined end time. Once the communication channel is established, the data source (IoV) will start sending messages one after another, while keeping the connection alive. *"Will it be a feasible solution to communicate with less cost?"* is one of the main questions that need to be discussed in the function establishment and task specification stage of prototype design.

New networking architectures such as Long Range Wide Area Networks (LoRaWAN ®) are becoming popular among the IoT community due to low cost, low power consumption, high data rate, robustness, network capacity, security, and coverage range (Mishra, 2018; Santa *et al.*, 2019). LoRaWAN provides a star-of-star network topology. The sensors are connected to the gateway hosts, which relay messages to the central network server (LoRa Alliance, 2015).

LoRaWAN specification is developed and maintained by the LoRa Alliance, which is an open association of collaborating members. The LoRaWAN protocol specification is open to the industry, so innovation and usage are free.

Table 4.1 shows the comparison of some of the features of Low-Power Wide Area Networks (LPWAN). According to LoRa Alliance (2015), the LoRaWAN is designed to support the IoT devices. LoRa Alliance claims that LoRaWAN gives better and extended battery life of sensor devices as it consumes less power, a greater capacity for future devices, a wide range of coverage due to the frequency, and reduced cost. This supports the answer to the aforementioned question, and it is feasible to implement this system in the real world with a low setting and running cost.

Table 4.1: LoRaWAN vs other LPWANs (LoRa Alliance, 2015)

Feature	LoRaWAN	Narrow-Band	LTE Cat-1	LET Cat-M	NB-LTE
Modulation	SS Chirp	UNB/GFSK/BPSK	OFDMA	OFDMA	OFDMA
Rx Bandwidth	500-125KHz	100Hz	20MHz	20-1.4MHz	20KHz
Data rate	290bps-50Kbps	100bps	10Mbps	200kbps-1Mbps	~20Kbps
Max. # Messages/day	Unlimited	140	Unlimited	Unlimited	Unlimited
Max Output Power	20dBm	20dBm	23-46dBm	23-30dBm	20dBm
Link Budget	154dB	151dB	130dB	146dB	150dB
Battery Lifetime – 2000mAh	105 months	90 months	-	18 months	-
Power Efficiency	Very High	Very High	Low	Medium	Medium-High
Interference immunity	Very High	Low	Medium	Medium	Low
Coexistence	Yes	No	Yes	Yes	No
Security	Yes	No	Yes	Yes	Yes
Mobility/localisation	Yes	Limited mobility, No Localisation	Mobility	Mobility	Limited mobility, No Localisation

Security

The system must make sure that the message (VT data) received is from the correct vehicles and not been taped or tampered. The token-based authentication is used to reduce the security issues from the application point of view. In addition to that, LoRaWAN provides added security.

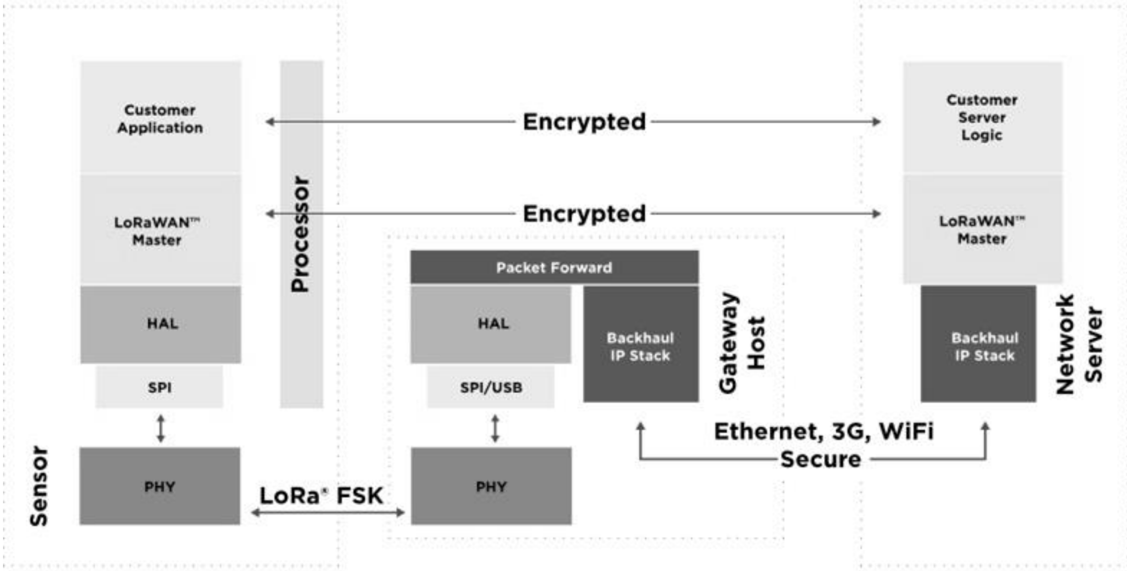


Figure 4.3: LoRaWAN Architecture (LoRa Alliance, 2019)

Figure 4.3 shows the LoRaWAN Architecture specification from the LoRa Alliance. The two-layer (Network level and Application level) end-to-end security is specified for IoT deployment. In layer one, a unique 128-bit Network Session Key is shared between the end-to-end device and the network server. A unique 128-bit Application Session Key is shared end-to-end at the application level. This enables a possible IoV application development concerning security issues.

Load Handling

In addition to connectivity and security concerns, the cloud-native system needs to be able to handle huge data from IoV devices in the real world. As discussed in this section, the backend must be able to handle stream and batch data from numerous VT devices at a time.

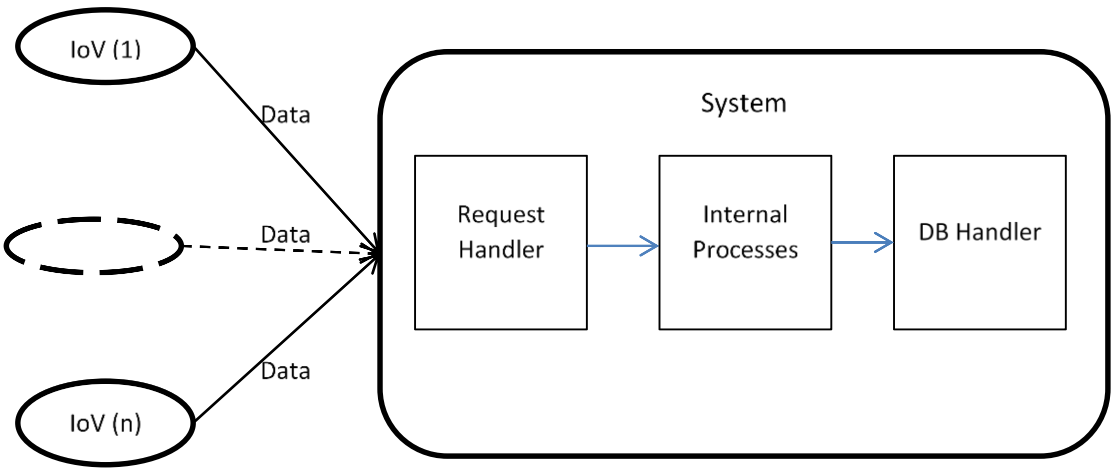


Figure 4.4: IoV and the system interaction

Figure 4.4 shows the interaction between the IoV data collection modules and the system comprising three main internal components, which in general, suffer from overloading or becomes unresponsive. A database handler process might process slower than the incoming request, or an internal process performing intense calculations may be slower than the incoming request, or the improper configuration may slow the process of the request handler. The system must be able to cater to each incoming request in real-time. A request may be a batch input or a data stream. Loss of this data may occur due to various reasons.

There are two common categories of server processes namely, stateful, and stateless. In stateful server process, the state of each service request is maintained. This is good for clients as it eliminates unnecessary resends. However, this adds more overload to the server, leading to delays in processing. As opposed to the stateful processes, stateless processes do not maintain any state histories of service requests, thus processes faster with less overload to the server, but has no recovery mechanisms.

Using stateful processes for a huge number of concurrent requests is not a good choice. However, even though the stateless processes are slim and work faster, it has its own problems. Loss of service request due to overflow or interruption may lead to the complete loss of data (request).

Backpressure is one of the closed-loop congestion control mechanisms in computer networks in which a congested node stops receiving data from the immediate upstream node or nodes. Backpressure is a node-to-node congestion control that starts with a node and propagates, in the opposite direction of data flow, to the source. The same technique can be practised between IoV devices (source) and the system (sink). When the system suffers from overload, it may reduce the rate of acceptance of the request from the source. However, here, when the source starts streaming, it would be a bit harder for the source to detect the backpressure and resend the packets again. This was one of the major concerns while designing the system architecture.

“Today applications are deployed on everything from mobile devices to cloud-based clusters running thousands of multi-core processors. Users expect millisecond response times and 100% uptime. Data is measured in petabytes. Today's demands are simply not met by yesterday’s software architectures” Jonas Bonér *et al.* (2014).

Jonas Bonér *et al.* (2014) believe a comprehensible approach to systems architecture is needed, and all necessary aspects are already recognised individually: they wanted systems that are Responsive, Resilient, Elastic and Message-Driven. They called such systems as Reactive Systems.

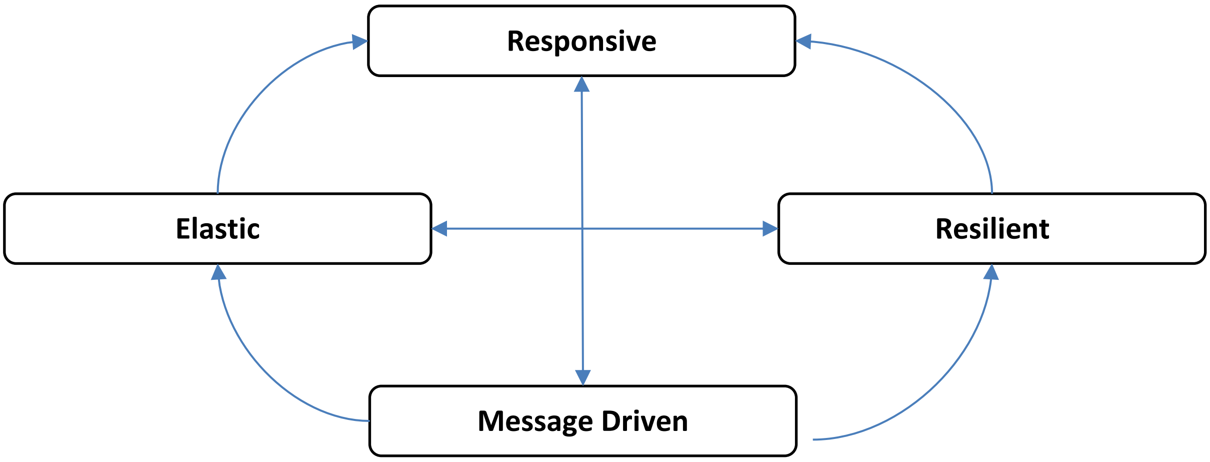


Figure 4.5: The Reactive Manifesto (Jonas Bonér et al., 2014)

Figure 4.5 shows the reactive manifesto by Jonas Bonér *et al.* (2014). In their manifesto, they declare that reactive systems are:

- **Responsive:** The high availability of the system which can respond to the user in a timely fashion is essential. The responsive systems focus on providing rapid and consistent response times to deliver the quality of service. This consistent activity, in turn, simplifies error handling, builds trust, and encourages further user interaction. The WIM application was built to ensure maximum uptime and minimal latency. The main priority was given to stream handling on the incoming VT data.

- **Resilient:** The unresponsiveness is mainly caused by a system failure; any system that is not resilient will be unresponsive after a failure. Resilience can be achieved by replication, containment, isolation and delegation. Isolating individual modules (components) is one of the strategies to keep the system up even in case of any component failure. The recovery could be made easily by recovering the faulty component. The reactive manifesto states that the client must not be burdened (resend the request) for any system failures. The VT data collection module must not be asked to resend the stream or batch in case of failure or any delay.

- **Elastic:** The reactive system must not have any single point of failure or central bottleneck. The system must be able to cope up and cater to the increasing number of requests, hypothetically handle infinite requests. The elasticity must be achieved by scaling system hardware and software. This was achieved by containerising the application and deploying it in Kubernetes cluster with replicating GlusterFS (Selvaganesan and Liazudeen, 2016) file system. In addition to that, the Cassandra DBMS was used with replication factor three for resilient data.

- **Message Driven:** Asynchronous message-passing enables Reactive Systems to form a barrier between components. This ensures loose coupling, isolation and location transparency. The component failures are communicated as messages. The message passing mechanism enables load management, elasticity, and flow control. Message passing also enables the application of backpressure when needed. Non-blocking communication allows recipients to only consume resources while active, leading to less system overhead (Jonas Bonér *et al.*, 2014).

Apache™ Kafka® is one of the widely used open-source stream processing platforms (Hiraman, Viresh and Abhijeet, 2018). Kafka uses Huge Persistent buffer for the bursts, does load distribution to a very large number of nodes, and enables horizontal scalability. Wampler (2018) suggests that standalone services like Apache Spark and Flink are better for batch processing, while the two libraries, Kafka Streams, and Akka Streams are good for streaming applications. Wampler also mentioned that both streaming libraries Kafka Streams, and Akka Streams, provide single-event processing with very low latency and high throughput. Kafka guarantees the message delivery by providing three message delivery semantics, *at most once, at least once*, and *exactly once*. Where in *at most once*, the messages may be lost but are never redelivered. In *at least once*, messages are never lost but may be redelivered. And in *exactly once*, each message is delivered once and only once.

Akka Streams also implements the Reactive Streams specification, which is built on top of Akka actor framework. Akka Streams is a simple, and powerful standard for defining composable streams. It, by default, uses the backpressure flow control mechanism. This keeps the internal stream “segments” robust against data loss and allows strategic decisions to be made at the entry points for the assembly, where it is more likely that a good strategy can be defined and implemented (Wampler, 2018). In here the reactive manifesto could be adhered in this system by using Asynchronous HTTP client, which is non-blocking and consumes no threads in waiting. By integrating with Akka streams for high parallelism, low resource solution, we could eliminate the complexities of dealing with multiple VT data streams, by using Akka stream in between the WIM application processes and Kafka Cluster for incoming data from the VT devices.

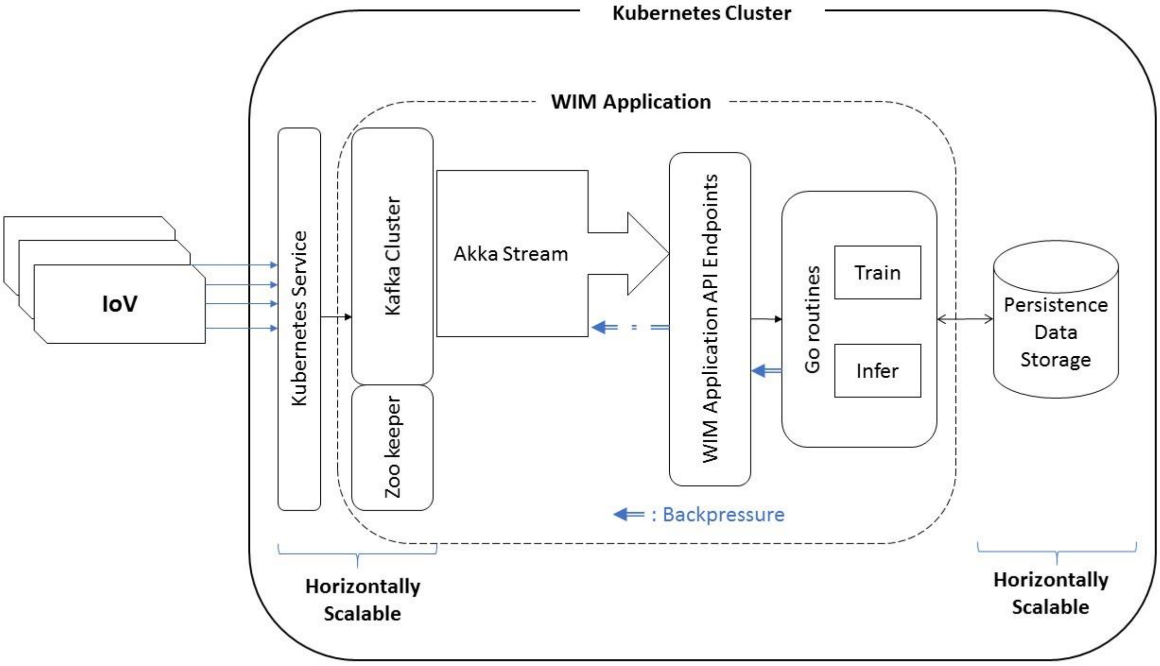


Figure 4.6: Kafka Fast data (streaming) architecture used in this research

Figure 4.6 shows the architecture of the prototype to handle fast data handling from multiple IoV devices. The prototype system is deployed on a Kubernetes cluster of five physical nodes (discussed in Section 4.2.2). The system receives tons of VT data from each IoV devices. The incoming VT data will be handled by a Kubernetes service. The Kubernetes service will then send the VT data to the available node, pod running the prototype WIM application. Prototype WIM application can handle fast streaming data by using Kafka Cluster and Akka Stream. The Kafka Cluster consumes and holds VT data to be ingested by Goroutines. Akka stream is used to stream each VT data (Kafka topics) for further processing. "Exactly once" delivery semantics was used in streaming. The Goroutines will process the VT data routed from its internal API endpoints. Persistence storage is used to store processed data (models, logs, events, results). This persistence storage is scalable horizontally to serve more data. The data overflow is handled internally without limiting (or requesting) the IoV devices to reduce the transmission rate (or resend). The generation of backpressure starts from the goroutines in case of any delay in processing. The backpressure is then propagated through the Akka streams to the Kafka Cluster. This will trigger the streaming to be flexible with the backpressure by reducing the data rate. But this leads the Kafka Cluster overflowed by the fast-incoming VT data accumulation. In such cases, the Kafka Cluster with the help of Zookeeper could scale-up horizontally.

4.2.2 Deployed Environment

Containerised applications are becoming a trend in this cloud era. A container is an application bundled with all its necessary components to run. It allows developers to package and isolate applications with their runtime environment, that is with all the files required to run. Kubernetes is a container orchestration engine which runs and manages Linux containers. Kubernetes is an opensource platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure (Oh, 2018). The application was containerised and deployed in Kubernetes cluster with five physical nodes, as shown in Figure 4.7.

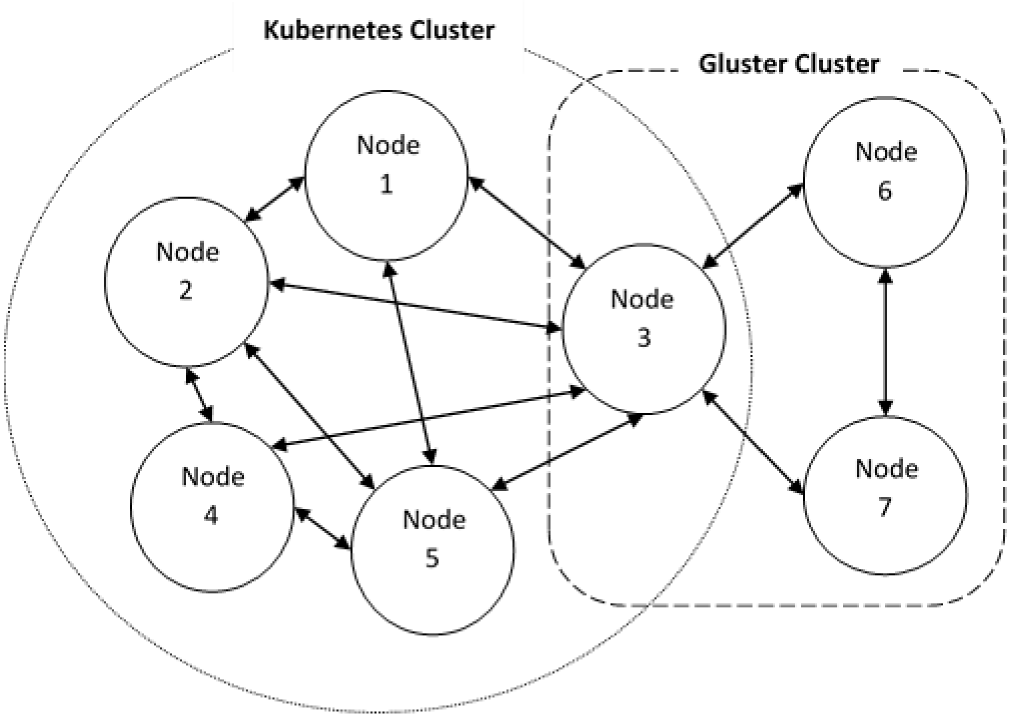


Figure 4.7: Arrangement of Kubernetes and Gluster nodes

Figure 4.7 shows the arrangement of seven nodes (PC's) used to set up the deployment platform. The five Kubernetes nodes were used. Persistence storage was provided by the three Gluster nodes, including one Kubernetes node. Kubernetes container orchestration engine runs several stateful applications, where the state of such applications is saved frequently. If a node dies or stops due to an unexpected event, then the Kubernetes will spin it off from the saved states. There are several persistent storage volumes that can be used in Kubernetes cluster to respawn and resume any stateful process. In this research, GlusterFs is used to maintain the persistence volumes for the Kubernetes Cluster.

GlusterFS is a distributed, software-defined filesystem where storage devices, called “bricks”, are selected on one or more nodes to form logical storage volumes across a Gluster cluster (Selvaganesan and Liazudeen, 2016; Gluster, 2019). It is easy to increase storage by simply adding more nodes, provides features like cross-node and cross-site replication, usage balancing, and iSCSI storage access (Gluster, 2019). Replicated GlusterFS volume architecture was used in this Gluster Cluster. This was done to overcome the data loss problem faced in the distributed volume. Exact copies of the data are maintained on all bricks. The number of replicas in the volume can be decided by the client while creating the volume. Three bricks were used to create a volume of 3 replicas. One significant advantage of such a volume is that even if one brick fails, the data can still be accessed from its replicated bricks. This volume is used for better reliability and data redundancy.

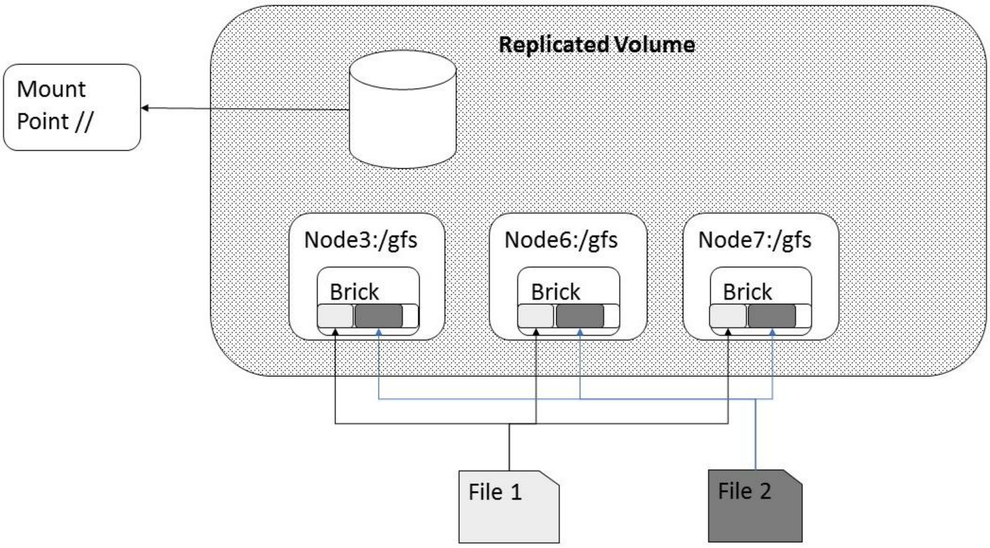


Figure 4.8: Replicated Volume Gluster Cluster Architecture

Figure 4.8 shows the implemented GlusterFS replicated volume architecture with three nodes. This replicated volume cluster was used to serve the containerised applications deployed on the Kubernetes by provisioning persistence storage.

4.2.3 Languages

The application was developed in Golang, also known as Go, which is a relatively new programming language developed by Google to construct its backend software services. Go programmes are compiled into native machine code. The compiler builds the binary for the native platform to run, which makes Go a cross-compiling to produce native binaries. Golang is perfect for microservice architecture, where the responsibilities of the application will be shared between similar services. Go was developed in the cloud computing era. Thus, it was built with modern software technologies in mind, especially containers. Go is one of the pioneers in container technologies. The famous container “Docker” was written in Golang (Mina Andrawos and Martin Helmich, 2017). One of the main advantages of Golang over the other high-level languages is that Golang can handle and make use of all the CPU cores, much like C++. This makes concurrent applications run much faster than other applications. Golang was chosen as the programming language due to reasons as mentioned above. In addition to Go, Scala was used to build the Akka Streaming endpoints consumed by the ML module developed in Go. R language was used to select the models; those models were then implemented in the application using Golang. R is one of the most popular and widely used for statistics, data mining and ML. R has rich ML packages and the support to Cluster integration.

4.2.4 Database management system

The Apache Cassandra is a Linear Scalable, fault-tolerant database management system to run on a commodity of hardware or cloud infrastructure. The Apache Cassandra NoSQL Database Management system was also deployed in the same Kubernetes cluster. The database is being used to store the weather information from a scheduled (corn job), retrieve stored VT data and weather data to train and test ML models, and to store the inferred output.

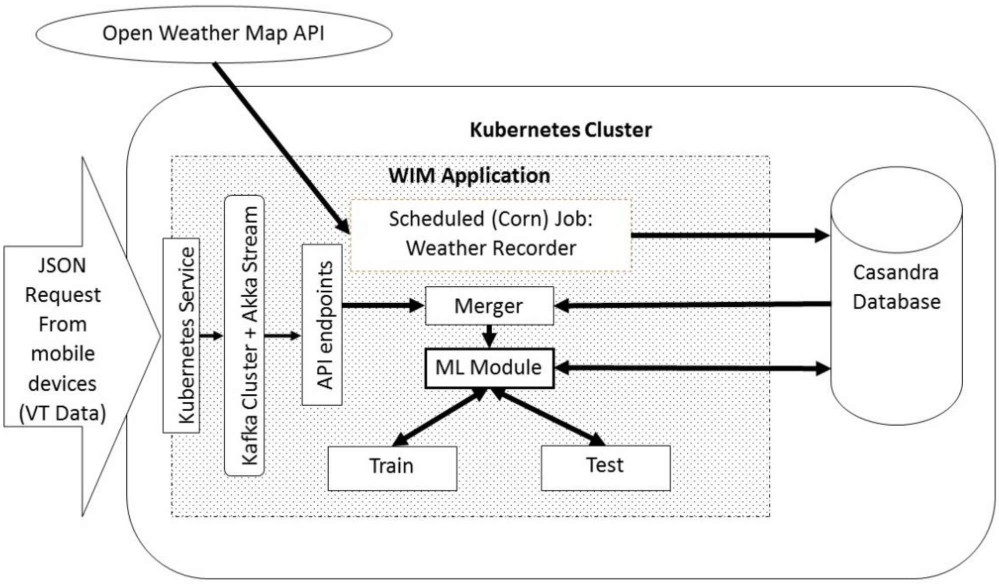


Figure 4.9: High-level system architecture of the prototype WIM application

Figure 4.9 portrays the high-level application architecture of the developed conceptual framework focusing on VT data ingestion. The WIM application has two main jobs namely, a scheduled weather data recorder and an ML application. The scheduled (cron) job (i.e. weather data recorder) reads the current weather information of the selected places from the OpenWeatherMap. The read weather data was then written into the Cassandra database for future use. This was done due to two reasons. The first reason was that the VT data might not be streamed real-time due to the unavailability of connectivity and other reasons, so fetching the current weather data at the time of receiving the VT data may not yield correct weather data. The second reason was that the limitation on API calls since we have used a free account for OpenWeatherMap API requests, the maximum API calls per minute was 60, and the total threshold was 7200. The OpenWeatherMap provides weather information for some specific fixed locations; for example, weather data were given for overall cities, not for fine locations. Keeping that in mind, the data was collected for known places where the vehicle was driven to collect VT data. The scheduled job (corn job) automatically collects current weather data of the prior set locations from the OpenWeatherMap and stores it into the Cassandra database.

The ML module was designed to infer the weight of a vehicle carrying (payload). ML models were chosen by training and testing with the data present in the backend. The WIM application was designed to use APIs to receive from a wide range of sources using JSON format. The WIM application can be accessed using the API endpoints on the ports exposed by the Kubernetes service. When a JSON post request hits the Kubernetes cluster, the service will map it to the specific node based on its availability. If the request is for training, then the merger application will store the incoming data into the database and triggers a Goroutine to merge the existing weather data with the VT data based on time value (timestamp). Similarly, the inference is performed when the VT data arrives at the correct API endpoint for inferencing. The inferred data is saved into the Casandra database. The saved results (inferred values with location) could be served to any frontend using a separate API developed in Golang.

4.3 Prototype System

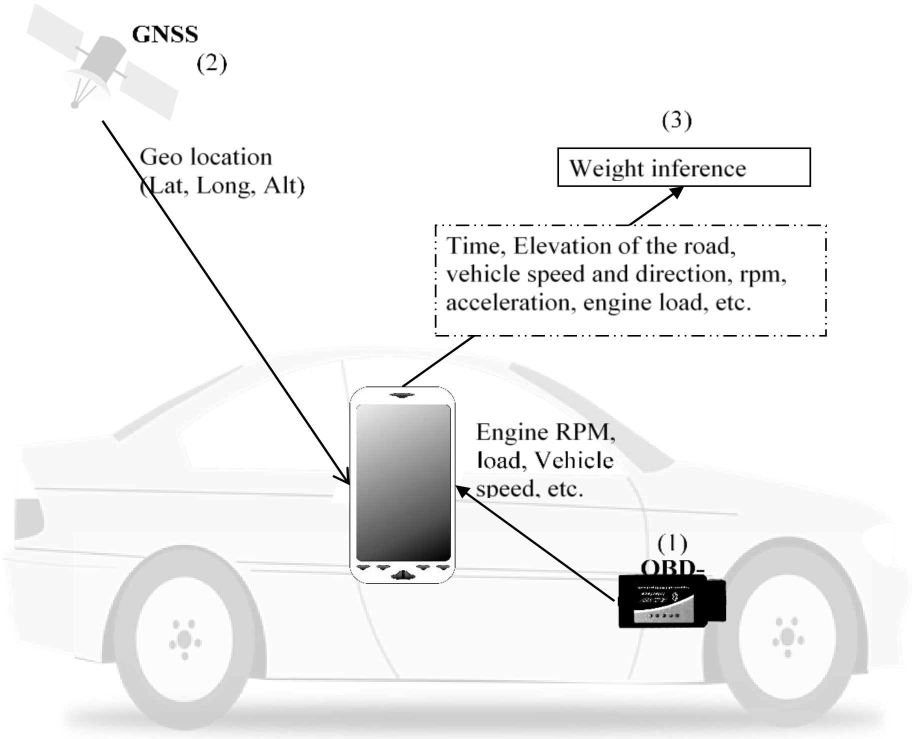


Figure 4.10: Representation of the developed system prototype with its components

A prototype of the system was developed to validate the idea of the new WIM system. The system comprises three components, which are:

- an OBD-II Bluetooth/WiFi module,
- an android mobile device,
- a WIM inference engine application is running on a Kubernetes cluster (discussed above).

The OBD-II module was used to collect the CAN bus data (i.e., Engine Control Unit (ECU) Data). The Android mobile device was used to fetch CAN bus data from the OBD-II module via Bluetooth or WiFi. Android mobile device collected ECU data for each second and stored it along with the data from the built-in GNNS position data. The collected/stored data was then sent to WIM API server using the REST clients through the service API endpoints. Figure 4.10 shows the schematic diagram of the developed prototype. An Android phone collected the data from the OBD-II module (1) via Bluetooth, its internal IMU, and GNSS (2). The collected data was then transferred from the phone to the back-end server WIM application (3). As mentioned in Section 4.1, the system was built to collect weather data from OpenWeatherMap API. The system collected weather data, including wind speed, wind direction, atmosphere temperature, atmosphere pressure, and humidity.

4.4 Data Collection

According to Wieringa (2016), the implementation of an artefact from idea to practice must start from small laboratory conditions, i.e. start development and test on the context of a specific group then move to the road credibility to test on many groups. The major goal of this research was to verify the idea of using VT and ML for WIM. The verification of this idea was done considering the context of a small car. The validation of these systems is yet to be done. The fully internal combustion engine, hybrid (electric + internal combustion engine), and fully electric motor are the available three different driving sources of the present-day vehicles. Internal combustion engine vehicles on the current market have the combination of features given in Table 4.2. Table 4.2 lists some of the features in internal combustion engine vehicles. A car having a combination of the features was used to verify the concept.

Table 4.2: Some features of internal combustion engine-powered vehicles

Feature	Values
Body Type	Car (Sedan/Coupe, Hatchback, Wagon)
Utility Type	SUV, ATV, MPV
Engine	
Capacity	600 cc – 5,000+ cc
Number of cylinders	2,3,4,6,8,12,14,16,18
Valves per cylinder	2 - 8
Alignment	Inline, V, Boxer, Rotary
Fuel Type	Gasoline, Diesel, LPG
Air intake	Turbocharged (exhaust/electric driven), or no turbocharger
Transmission	
Auto	Hydraulic Auto Transmission, Continuous Variable Transmission, Dual Clutch Transmission, Automated Manual Transmission.
Manual	Forward Gears - 4,5,6,7
Drive	2 Wheel (Front/Rear), 4 Wheel

The data collection was done on a Ford Fiesta manufactured in the year 2015, which is a 1.4l four-cylinder gasoline engine with the front-wheel-drive with five manual transmissions and the curb weight of 1110Kg. Torque Lite Application on an Android Mobile phone running Android OS 8 was used to collect the data from the ELM 237 OBD Bluetooth Scanning device. The car was driven in controlled and uncontrolled environments. The controlled data collection was done at Cape Peninsula University of Technology (CPUT) premises shown in Figure 4.11. Volunteers weighing different weights participated as passengers during data collection. The car was driven only on first and second gears.

Figure 4.11: Controlled data collection road track on google map

The Landscape of CPUT contains inclines (up to 40-degrees) and low (0-degree elevation) roads. The controlled data collection was done on sunny days with wind no more than 5km/h. The uncontrolled data was collected from the daily commuting of car for four different days with a similar weather condition.

VT data was labelled with the total carrying weight, also known as payload (i.e., the sum of the masses of the passengers and the diver with the mass of any bags carried). Since the density of the fuel is 0.7kg/l, and the fuel tank capacity is 43 litres, it makes a significant 30kg difference in total weight. The weight of the fuel was also considered in four-quarter blocks by observing the fuel gauge reading.

4.4.1 Data

Various data sets were collected from the OBD-II dongle, smartphone, and the OpenWeatherMap's Weather API. The data collection application logged the data for every 1-second interval (sample rate = 1Hz).

Table 4.3: Details of the variables collected during the data collection

Variable	Description	Source	Units
ACC	Acceleration	$=(\Delta VS*0.27)/\Delta t$	ms^{-2}
ALT	Altitude (Meters Above Sea Level)	GNSS	m
LAT	Latitude	GNSS	-
LON	Longitude	GNSS	-
ELE	Elevation	$=\arctan(\Delta LAT/\Delta DD)$	degrees
VS	Vehicle Speed	OBD/GNSS	kmph
DD	Drive Distance	OBD	km
EL	Calculated Engine Load	OBD	%
RPM	Revolutions per Minute	OBD	r/min
TP	Throttle Position	OBD	%
HUM	Humidity	OpenWeatherMap	%
TEM	Temperature	OpenWeatherMap	K
PRE	Atmospheric pressure	OpenWeatherMap	hPa
WS	Wind speed	OpenWeatherMap	kmph
WD	Wind direction	OpenWeatherMap	degrees
Weight	Payload	User input	kg

Table 4.3 shows the details of the data collected from different sources during the initial data collection. ECU data such as Vehicle Speed (VS), Throttle Position (TP), Engine RPM (RPM), Calculated Engine Load (EL), and Drive Distance (DD) were collected from the OBDII device. The global position data such as Latitude (LAT), Longitude (LON), Altitude (ALT) were collected from the smartphone's GNSS unit. The combined data with the timestamp and the geolocation was then used to extract the weather information form the stored weather database. Since EL depends on airflow, standard temperature and pressure, those readings were not recorded nor included in the feature set to reduce multicollinearity.

4.5 Correctness of Data

4.5.1 Weather data

Wind direction data from OpenWeatherMap API consists of the wind speed and the wind direction in meteorological degrees. The wind speed and direction directly influence the driving force of a vehicle. Thus, it is an essential data for the inference system.

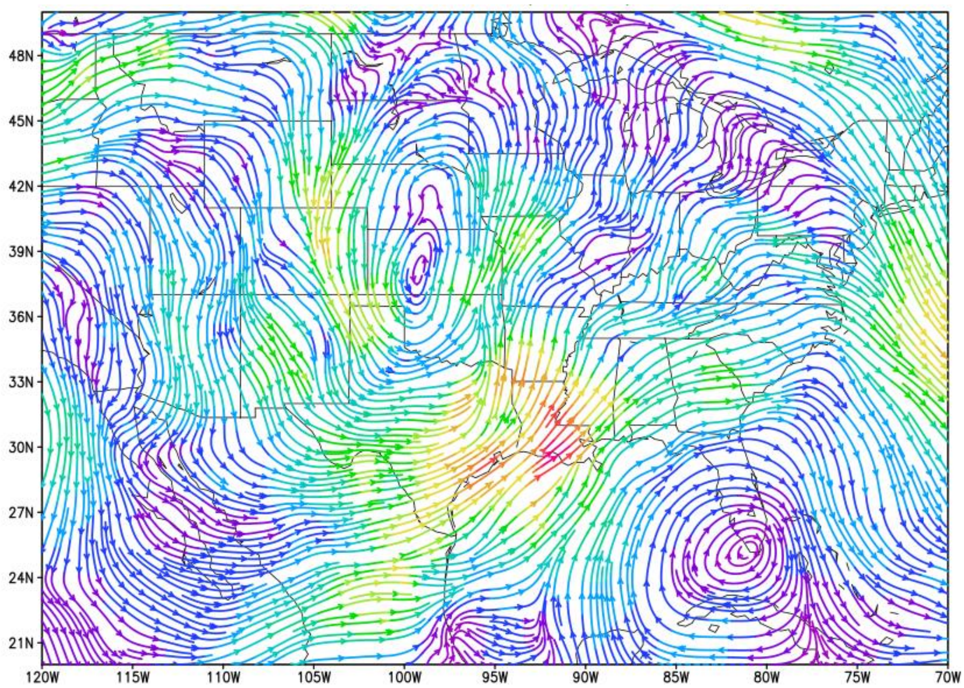
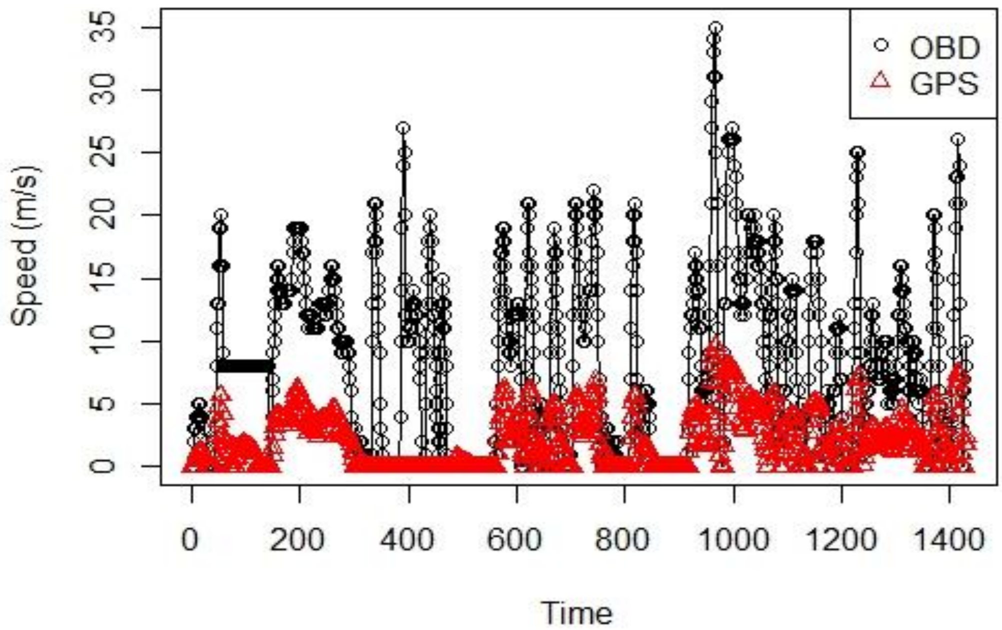


Figure 4.12: Sample wind flow direction map from Natural Environment Research Council (NERC)

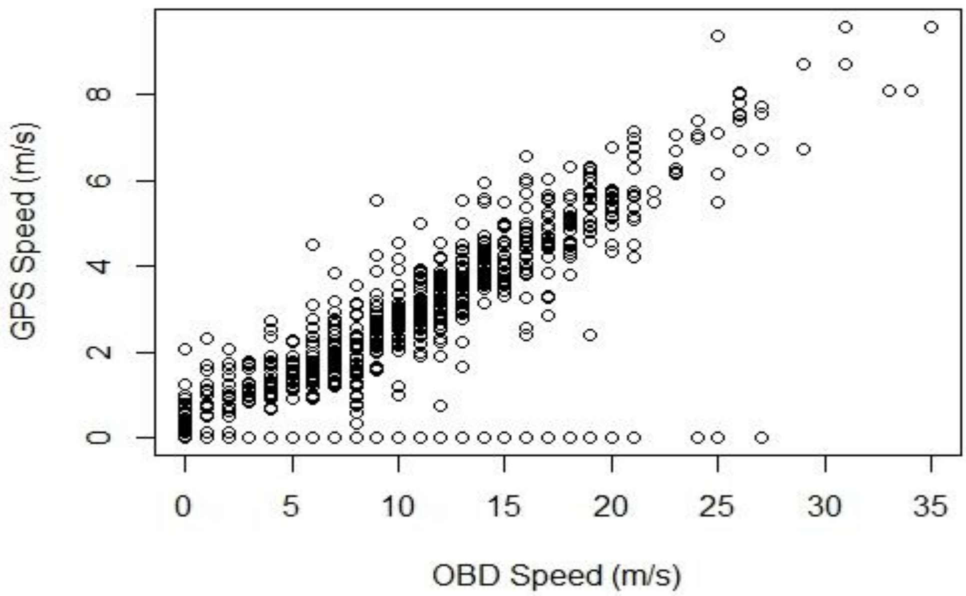
Unfortunately, our current ability to monitor the weather and environmental conditions is still severely limited in both time and space. The weather data available now are with spatial granularity in the order of several square kilometres, and time resolution in the order of 1 h (Massaro *et al.*, 2016). Figure 4.12 shows us that the flow of wind (wind direction) will not be the same at all places in an area. The direction of the wind and the speed may vary due to the landscape and the objects. The resolution of the weather data obtained was two-hour. Most of the data recorded have remained unchanged, or data with minimal variance, and the wind direction and wind speed data need to be instantaneous at each location where we collect VT data. The model errors were higher with the weather data incorporated. Thus, weather data was excluded while selecting models in this research.

4.5.2 Speed from ODB vs GNSS

The vehicle speed collected from the OBD vs GNSS is shown in Figure 4.13. Pearson's product-moment correlation coefficient (PPMCC) (Swinscow, 1997) was used to check the correlation between the two different readings.



a



b

Figure 4.13: Difference in speed reading from OBD and GPS

PPMCC between two vectors $X = \{x_1, .., x_N\}$, and $Y = \{y_1, .., y_N\}$ is,

$$PPMCC = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}$$

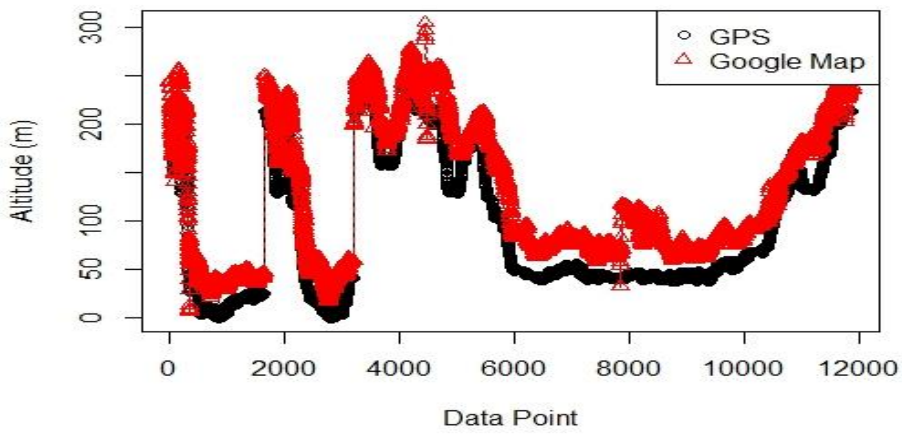
Equation 4.1

where,

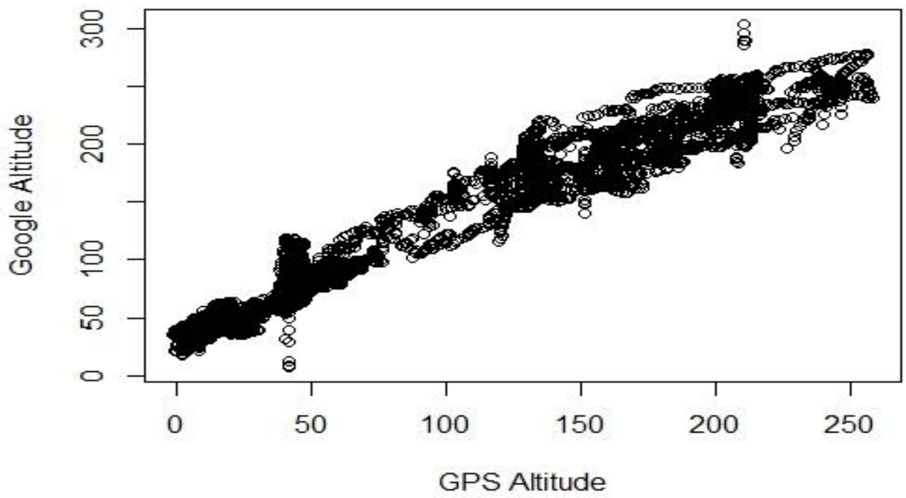
$S_{xx} = \sum_{n=1}^{N}(\bar{x} - x_n)^2$, $S_{xy} = \sum_{n=1}^{N}(\bar{x} - x_n)(\bar{y} - y_n)$

PPMCC of the speed readings from OBD and GNSS is 0.842. The zero readings for nonzero values of Speed OBD readings are due to the time taken to fix GNNS satellites for positioning. Due to this reason, the speed in this research was chosen from OBD reading.

4.5.3 Altitude form GNSS vs Google Map API



a



b

Figure 4.14: Difference in altitude from GPS and google map for un-controlled driving data. (a) Altitude readings, (b) Correlation graph

Altitudes from Google Map API for a list of latitude and longitude position brings a different value from GNSS sensor altitude, as shown in Figure 4.14. The PPMCC between these two altitude measurements is 0.976.

4.5.4 Road Gradient (Elevation angle)

The phone's rotation sensor was tested to be used to find the elevation angle of the road. In order to obtain the elevation angle, the phone was rigidly placed parallel to the chassis of the vehicle assuming the vehicle chassis will always be parallel to the road surface. Due to the suspension system of the vehicle, the nose lift and nose down happened during the acceleration and braking. Similarly, the linear acceleration calculated from

IMU was not enough to capture the lateral acceleration/deceleration (ACC) of the vehicle due to throttling and braking. Following equations, Equation 4.2 and Equation 4.3 were used to calculate ACC and ELE, respectively .

$$ACC = \frac{\Delta VS \times 1000}{\Delta t \times 60 \times 60} \ \ ms^{-2}$$

Equation 4.2

Where ΔVS is the change of vehicle speed in kmph, Δt = change of time in seconds. Road Gradient/Elevation Angle in degrees:

$$ELE = \tan^{-1}\left(\frac{\Delta ALT}{\Delta DD \times 1000}\right) \circ$$

Equation 4.3

Where, ΔALT is a change of altitude in m, ΔDD=drive distance in km.

4.5.5 Rate of Data collection

Systematic measurement errors such as lag time and hysteresis may be present while reading the values from CAN bus data and GNSS; these errors are very hard to detect and eliminate.

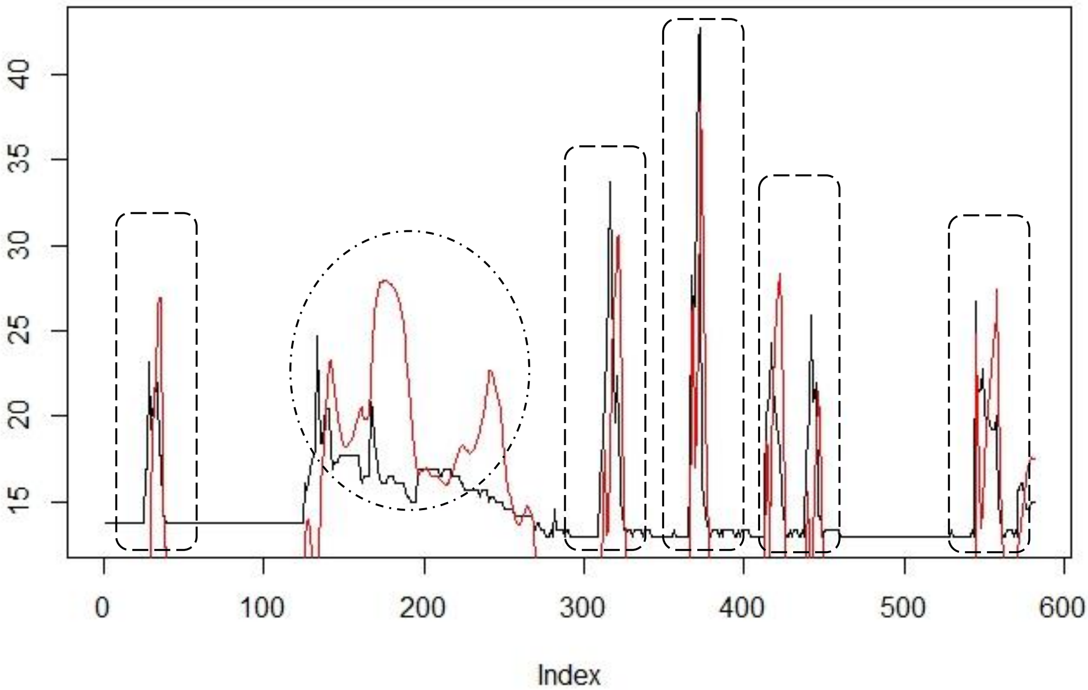


Figure 4.15: Throttle response time (Throttle position and RPM)

In Figure 4.15, the graph shows the response of the engine RPM (denoted in red line) to the throttle input (denoted in black line). When there is a change in the throttle position that change reflects in the engine RPM. The parts of the graph in rounded rectangles show the delay in engine response during the throttle change in normal conditions, which is when either the clutch is engaged (pedal released) and accelerating, or the clutch is disengaged. The delay in engine response in those regions is clearly visible. It was found that there is a 0.6second delay in average between peaks on input and its response. The area denoted by the oval shows the reverse response (negative or irregular response) of the engine. This was due to the engine braking, that is when we deaccelerate by reducing throttle while the clutch is engaged. Such data was considered inappropriate and explained in the next section.

The frequency of parameters ranged from 1Hz to 100Hz but was collected at the rate of 1Hz. The reduced rate of data collection might have missed some crucial facts from that data.

4.6 Data Pre-processing

The correctness of the data influences the model accuracy. The model needs to be trained with carefully chosen data for better and robust accuracy. The data from the start of a journey to the end was plotted to observe the behaviour of independent variables. The graph in Figure 4.16 shows the values of RPM, VS, EL, and ACC of a journey for Point A to Point B within the 20s.

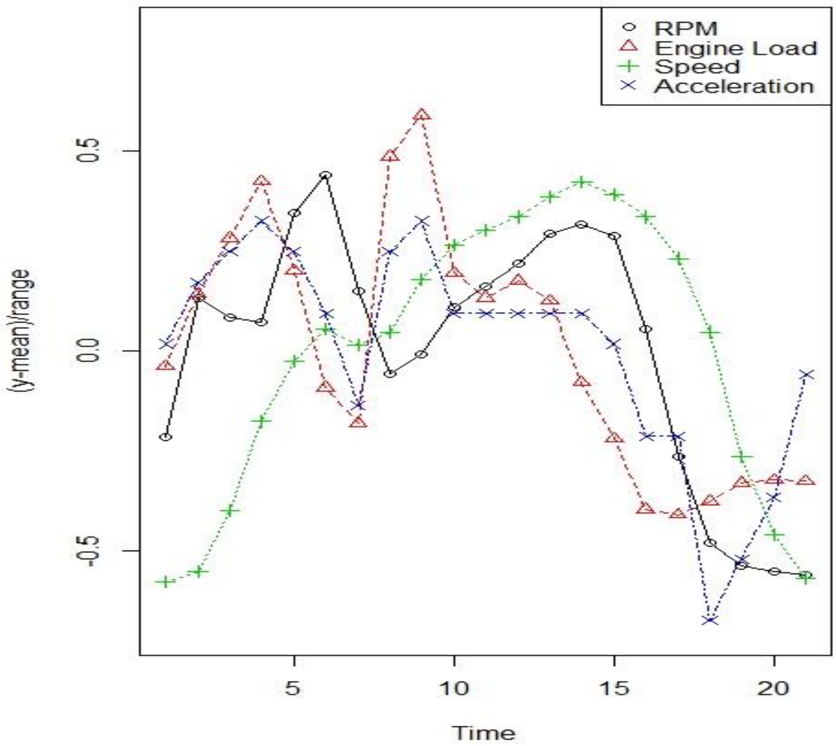


Figure 4.16: RPM, Elevation, Speed, Acceleration for a drive from point A to point B

The first spike on the EL shows the gear change from first gear to second gear. During this period the clutch will be released to separate the engine and transmission, and throttle position will be decreased. Thus the RPM will also be reduced. This speed difference (i.e. ΔVS) is very lean; therefore, the acceleration reaches zero, then shoots up when the gear is changed. The graph segment between time greater than 15 depicts the braking (deacceleration) event to bring the vehicle to a stationary state.

Acceleration may occur due to two different reasons; 1) vehicle is on an inclined or flat surface (i.e., ELE>=0) when TP is high, RPM is high, and EL is high. 2) Vehicle is on a declined surface (i.e., ELE<0) and the TP, RPM, and EL are low, where the vehicle starts moving due to the gravitational pulling force.

Similarly, the deacceleration without applying brake can occur due to two different conditions; 1) on an inclined surface (i.e., ELE >0), low TP, low RPM, and low EL . 2) on a flat or declined surface (i.e., ELE<=0) high RPM, low TP, and low EL (usually on low gears) as explained by the oval shape in Figure 4.15.

Training ML models with this complex and noise data did not yield a good model accuracy. Consequently, the model is trained with data points where (ACC>=0 & ELE>=0 & RPM > Minimum RPM & TP > Minimum TP).

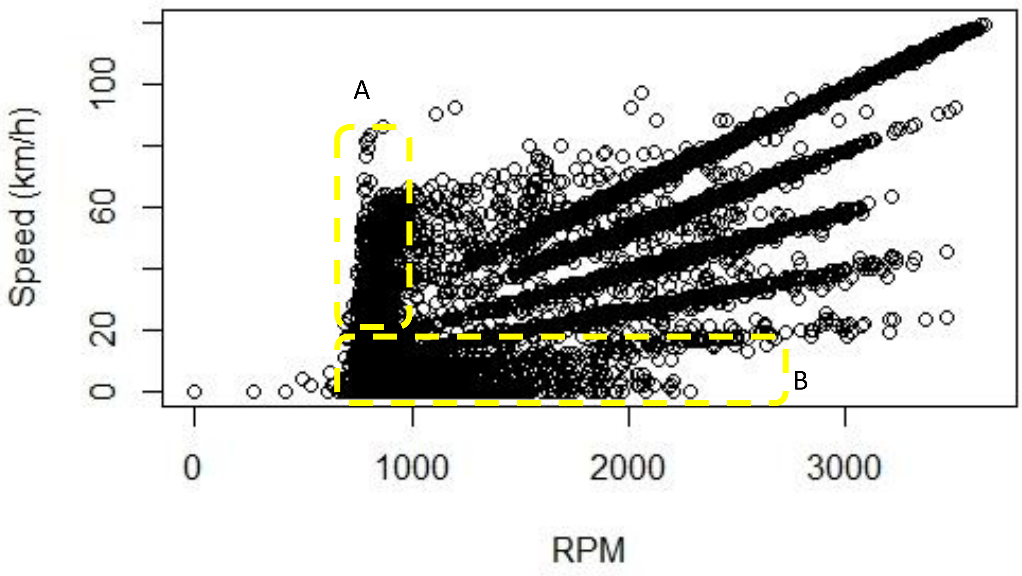


Figure 4.17: RPM vs Speed

Figure 4.17 shows the RPM vs Speed graph for the uncontrolled data collected on different payloads (95, 110, 112, 180, 240, and 320kg). This graph clearly shows the correlation between Speed and RPM for different gears. It is easy to distinguish the five gears which are represented by five slope lines in the graph. Decreased TP and RPM cause (region A in this graph) during gear changes and braking. Region B denotes our interesting area in this graph, where 0<VS<20. Region B contains the data obtained when the vehicle changed its state from stationary to moving. The speed gain during the first gear was captured for different payload settings. The left corner of this region B is denser than other regions in this graph. This makes us focus on this region as other gear settings do not show any significant patterns for different payloads.

4.7 Data Transmission

The collected VT data must be sent to the WIM system either in batch or stream fashion. Streams with short bursts would be preferable than long burst streams. Assume a vehicle data collection device (sender) collects the VT data at a rate of 1 Hz. And it starts sending or queuing its' VT data to the system from the start of the journey. In such situations, the volume of data throughout the journey depends on the duration of the journey. The amount of data for the VT devices to store should be minimised for a better reactive system. Further, we have noticed that not all VT data is useful for inferring the weight.

The following steps explain the data collection process deployed in the prototype system. In here, Speed is the current speed of the vehicle. Vehicle identifier (VID) is a unique identifier assigned to each vehicle. The route identifier (route ID) is a combination of VID and start time.

```
On the Sender side:
if (0<speed)
  routeID ← VID + StartTime
  if (speed<20)
        if (connected to network)
              Stream VT data
        else
              Queue VT data
  else
        Log Locatiodata, routeID
else
  Connect and Send VT data queue, Log
End
```

The VT data device has two main functions namely, Streaming and Logging. The size of the VT data stream is reduced by limiting the VT stream data by only streaming during 0 - 20kmph speeds. By doing this, we reduce the streaming time as well as the accumulation of unnecessary data. If the VT device is connected to the backend, then, the data is streamed. Otherwise, the VT data is queued for streaming. At the backend, inferencing is done by the steamed data (during the drive form 0 – 20kmph) for each routeID. If the vehicle speed is greater than 20kmph, then the VT data collection device logs the geolocation (GNSS data) with the generated routeID for every second. The backend merges the inferred weight of a routeID with the logged data to track the payload throughout the journey of a vehicle. When the vehicle stops and starts again, then the new VT data is sent to infer the weight again. Each stop and go triggers the inferencing. This allows tracking any vehicles which overload at any point of their journey.

4.7.1 Data Extraction

Once the VT data is available on the system, we have to train each new vehicle to obtain the inference model. The following algorithm shows how the training data extraction works for each vehicle.

Input: Table IT contains values of ACC, VS, RPM, EL, ELE, TP for each 1s timestamp.

Output: Table OT contains values specific for training a model.

<u>Procedure</u>:

for each row *i* in IT,
  if ((0<VS<20) AND (ACC>=0) AND (ELE>=0) AND (RPM > Min (RPM))
  AND (TP > Min (TP)))
    Add row *i* into OT
  else
    go to next row *i*+1.

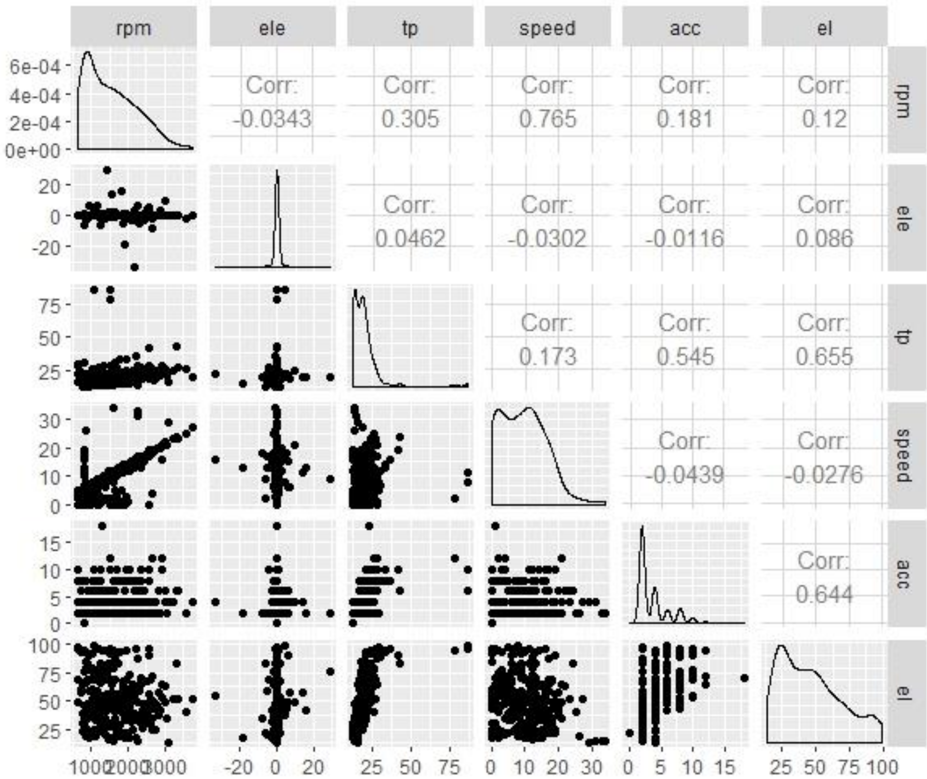


Figure 4.18: Scatter plot and Correlation matrix of the primary features from the dataset

Figure 4.18 describes the sample extracted data using the data extraction process. This extracted data was then used to choose the ML model. In this data, the correlation between VS and RPM is 0.76, significantly higher than other correlation values.

4.8 ML Model Selection

This research focused on regression models rather than classification models. No attempt has been made to test a classification model classifying overloaded and legally loaded vehicles. This was so as to not violate the laws and not damage the testing vehicles. On the other hand, an attempt has been made to test the weight inference system using regression models. A set of ML algorithms such as multiple linear regression, ANN, Decision Trees, and Bayesian regression were chosen to test. ANN performed better than other ML algorithms. Selection of features for each ML algorithm was based on their performance.

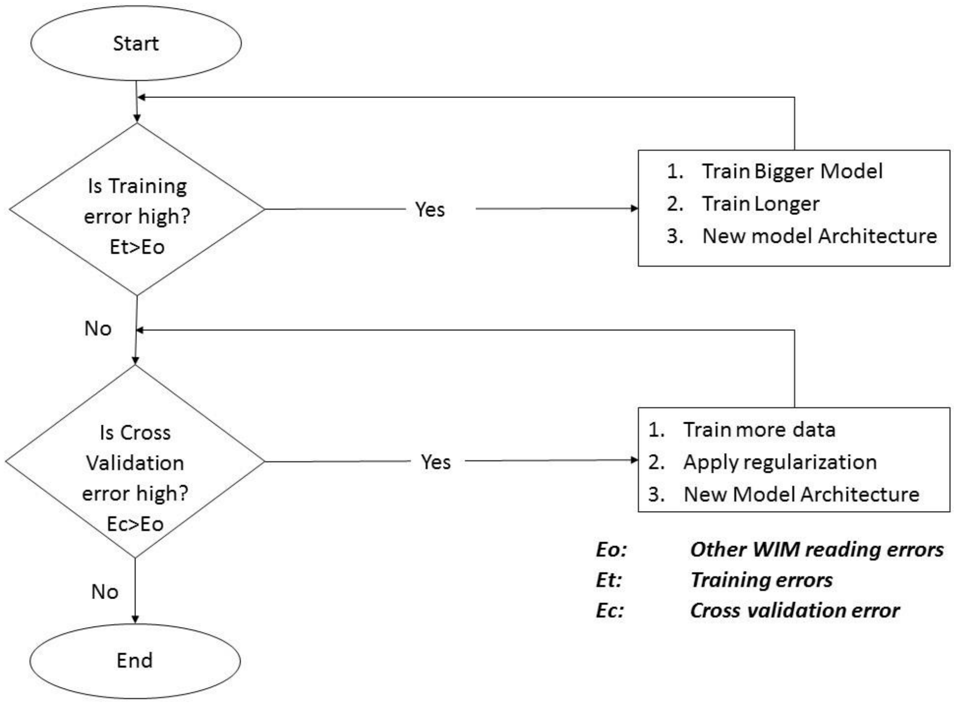


Figure 4.19: Flowchart of the ML model selection process

Figure 4.19 shows the ML model selection process carried out in this research where Eo = 5% - 35% with 95% confidence level was obtained from the literature (Gajda, Burnos and Sroka, 2018).

4.8.1 Feature Creation

Feature engineering is the most difficult and time-consuming part of ML projects (Domingos, 2012). The raw data we gathered was not in a form amenable to learning. This part of the research has consumed a considerable amount of time. After performing data pre-processing, the pre-processed data was then filtered using the data extraction process. The chosen data was then used to build Learning Models. The correlation matrix was then used to check the correlation between variables.

Table 4.4: Correlation Matrix of Base Features and dependent variable (weight)

	RPM	ELE	TP	VS	ACC	EL	Weight
RPM	1.00						
ELE	-0.03	1.00					
TP	0.30	0.04	1.00				
VS	0.76	-0.03	0.17	1.00			
ACC	0.18	-0.01	0.54	-0.04	1.00		
EL	0.11	0.08	0.65	-0.02	0.64	1.00	
Weight	0.05	0.09	-0.07	0.12	0.09	-0.03	1.00

Table 4.4 shows the correlation matrix between the collected features. The correlation matrix does not reveal any direct correlation between the base features and the dependent variable.

Correlation between the independent variables is known as multicollinearity. In here the VS and RPM are highly correlated with the value 0.76. RPM was removed in some settings to check the effect of removing multicollinearity. The reason for choosing RPM instead of VS because RPM is less correlated to weight (0.05) than VS (0.12). Some new features were added by multiplying existing features and finding the powers of selected features. ACC, VS, RPM, EL, ELE, TP are used to create new features using non-linear functions such as Log(x), Sqrt (x), and Power (x, -1), Power(x,2). Where Power (a, b) = $a^{\mathbf{b}}$. Feature crossing is also done to obtain new features by multiplying and dividing existing features.

4.8.2 Feature Selection

Selecting the best set of features is essential for the better performance of the ML model. Keeping a higher number of features may lead to many hazardous situations. The higher number of feature space makes the model harder to interpret. Space and time complexity will also be affected by the number of features. It could also lead to model overfitting in some cases. Handling higher dimensional data will also be an issue with higher feature space.

There are several methods available for feature selection. Stepwise regression, penalised regressions (i.e. Ridge, Lasso, and Elastic) and principal component based regression (Kassambara, 2018). According to Kassambara (2018), stepwise regression is ideal for high-dimensional data with multiple features. Stepwise regression was done to find the best number of features. The feature selection of stepwise regression uses Root Mean Squared Errors (RMSE) (Hocking, 1976). It showed that using the four-variable model results in the best RMSE value. A stepwise feature selection based on Akaike Information Criterion (AIC) (Hocking, 1976) was also performed. The results obtained using the stepwise regression is discussed detailed under Results and Discussion. Nine different settings were made, and the performance was measured based on their Residual Standard Error, Degree of Freedom, a p-value of the model, R-squared, and Adjusted R-squared.

The Following settings were done to choose the model.

1. Simple regression with all base features
2. Simple regression with all base features excluding RPM (due to multicollinearity)
3. Setting 1 with Single feature crossing (i.e. each base feature is multiplied with another)
4. Setting 2 with single feature crossing
5. Introducing new features by adding non-linear functions such as Sqrt (x_i), Power (x_i, -1), Power (x_i, 2), and Log (x_i) to setting 1
6. Introducing new features by adding non-linear functions such as Sqrt (x_i), Power (x_i, -1), Power (x_i, 2), and Log (x_i) to setting 3
7. Setting 6 with two feature crossing
8. Selecting the best features picked from 7 based on significance value
9. Feature Selection on Setting 6 using Stepwise AIC with 3 feature crossing

4.9 Evaluation

The ML models were selected by the evaluation based on several matrices, which are discussed separately.

4.9.1 Linear Regression

The linear regression models were evaluated based on the following conditions:

a) Linearity:

The relationship between the independent (explanatory) and the dependent (response) variable should be linear. This was tested using the residual plot.

b) Nearly normal residuals:

Residuals should be nearly normally distributed and centred at zero, presence of noise data (unusual observations) may not satisfy this condition. This was verified using normal Q-Q plot.

c) Constant variability (homoscedasticity):

Variability of points around the least-squares lines should be roughly constant. This was checked using the residuals plot.

d) Significance (p-value):

The p-value determines the overall significance of the model. If the p-value is smaller than 0.05, then the model may be considered significant.

e) R-squared Error:

The R-Squared error, R^2 of a regression model is:

$$R^2 = 1 - \frac{\sum_{n=1}^{N}(y_n - \hat{y}_n)^2}{\sum_{n=1}^{N}(y_n - \bar{y})^2}$$

Equation 4.4

Where, $\hat{y}_n$ is the n^th predicted value, $\bar{y}$ is the mean of the response values. The closer the value to 1, the more the points tends to fall along the regression line, thus, the stronger the linear relation the two sequences have. $R^2 = 1$ means the two sequences have perfect linear relation, while $R^2 = 0$ means they have no linear relation at all. Fitness of the model is measured with the R^2, Greater the R^2, better the model, i.e. model is very good if $R^2 = 1$, very poor if $R^2 = 0$ (Hansheng Lei and Govindaraju, 2004). Negative R-Squared values could result if the model fits very worst. But, R^2 alone cannot be used to evaluate the model performance, since overfitting models may also produce greater R^2 values.

f) Adjusted R-squared

Due to the limitation in R^2, adjusted R-squared, R^2_{adj}, is used in addition. Several formulas were proposed and are in use, among those we used McNemar's formula:

$$R^2_{adj} = 1 - \left[\left(\frac{N-1}{N-k-1}\right)(1 - R^2)\right]$$

Equation 4.5

Where N is the number of Observations, k is the number of independent variables (features).

g) Degrees of Freedom

Degreed of Freedom (DF) is the number of observations subtracted by the number of independent variables (Features).

$$DF = N - length(x_n)$$

Equation 4.6

DF makes sure that the data size is greater than the size of the feature set.

h) Residual Standard Error ($\hat{\sigma}$)

This is calculated from the Sum of Squared Error (Equation 2.7) and the Degrees of Freedom.

$$\hat{\sigma} = \sqrt{\frac{SSE}{DF}}$$

Equation 4.7

We can say the model would produce $\pm\hat{\sigma}$ error on average. Suppose the residuals are approximately normally distributed, then $\hat{\sigma}$ can be used to say that $\frac{2}{3}$ or 65% of the result is in the range of $\pm\hat{\sigma}$, and 95% of the prediction is in the range of $\pm 2\hat{\sigma}$.

4.9.2 Artificial Neural Network

ANN architectures with the different numbers of hidden layers were tested. The performance of the architecture was seen based on accuracy, R-Squared (Equation 4.4), and Adjusted R-squared (Equation 4.5). The accuracy was calculated as follows:

$$Accuracy = 1 - mean(\frac{|y - \hat{y}|}{y})$$

Equation 4.8

4.9.3 Bayesian Regression and Decision Trees

In addition to the multiple linear regression, Bayesian regression was also conducted to determine the posterior distribution of the independent variables. All the basic features were used in Bayesian regression and Decision Tree Algorithm. R-Squared, Adjusted R-Squared, Mean Squared Error, and Standard residuals were used to evaluate those models.

CHAPTER 5

DISCUSSION

In this chapter, we discuss the following:

Section 5.1. Evaluation of Model performance Regression model
Section 5.2. Evaluation of Bayesian Regression
Section 5.3. Evaluation of Decision Trees
Section 5.4. Evaluation of ANN Model performance
Section 5.5. Model performance
Section 5.6. WIM System Performance
Section 5.7. C-K theory in action

5.1 Regression Model performance

Table 5.1 summarises the model performance results obtained from the previously mentioned nine settings.

Table 5.1: Results of the 9 settings

Setting	Residual Std. Error	Degrees of Freedom	p-value	R-squared	Adjusted R-Squared
1	34.46	298	0.0232	0.047	0.0284
2	34.44	**299**	0.0161	0.045	0.0293
3	34.88	283	0.3899	0.073	0.0042
4	34.76	289	0.2503	0.059	0.0110
5	33.53	281	0.0021	0.149	0.0800
6	32.07	54	0.1839	0.850	0.1582
7	41.24	2	0.7507	**0.990**	-0.391
8	28.37	54	0.0165	0.883	0.3412
9	**23.1**	88	**6.322e-08**	0.8736	**0.5633**

Setting 1 shows that the model is significant, but the R-squared and adjusted R-squared values are significantly low. The degree of freedom is high due to the lesser number of features.

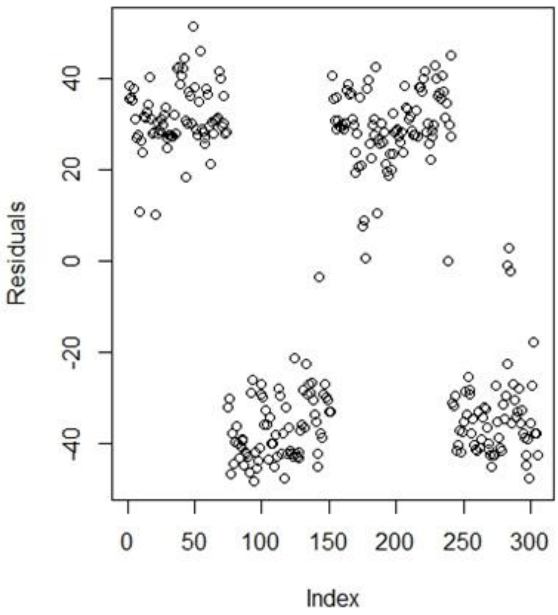


Figure 5.1: Residuals plot for Settings 2

Setting 2 shows a better result than Setting 1 with smaller p-value and adjusted R-squared; this is due to the removal of one feature from the previous setting. However, the residuals plots show the non-linear relationship between the independent variables and dependent variables. Figure 5.1 and Figure 5.2 show that the residuals reveal that the three conditions for linear regression, as stated in the previous chapter were not met. The new features were introduced by applying non-linear functions to the base features. This was tested with Settings 5 and above.

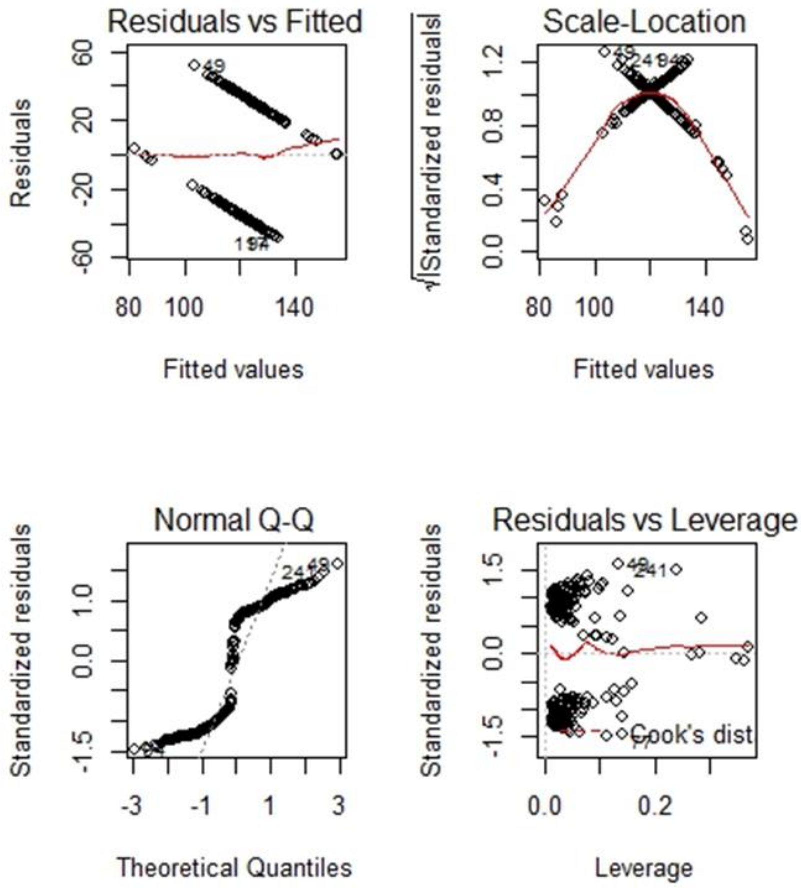
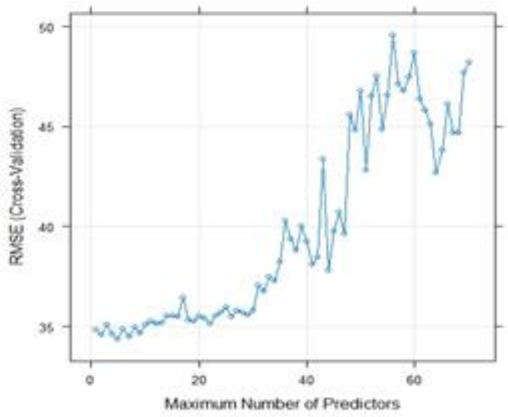


Figure 5.2: Settings 2 residuals plots

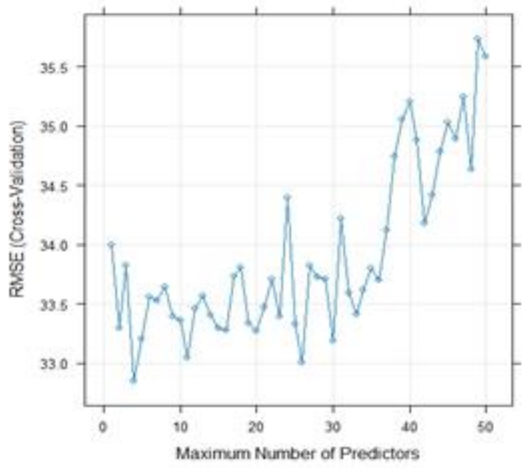
Settings 3 & 4 did not yield any better performance values than Settings 1 & 2. However, Setting 5 showed a significant improvement in performance with lesser p-value and higher R-squared and adjusted R-squared values; this again confirms that the features (independent variables) are non-linearly correlated to the dependent variable. Even though Setting 6 showed higher costs than previous settings, it is still weak due to the higher p-value.

Setting 7 yields a greater R-squared (mostly overfitted) with more significant p-value and a small degree of freedom. The negative value of adjusted R-squared reveals that the model is suffering from too many surplus features. It seems the number of features is higher than the number of observations in Setting 7.

Setting 8 is made by only choosing the significant features from Setting 7. This resulted in a decent result with significance, better R-squared and adjusted R-squared. The model is complex to interpret but performs better than the simpler models. Above all, the other Setting 9 with the triple feature crossing and using Stepwise AIC resulted in better results.



a



b

Figure 5.3: Stepwise regression feature selection using (a) 10-fold Cross-validation and (b) Leave-one-out Cross-validation

Stepwise linear regression feature selection based on setting 6 resulted in graphs, as shown in Figure 5.3. The graph (a) shows 10-fold cross-validation results and the graph (b) shows leave-one-out (i.e. k=n) cross-validation. Both graphs show the best tune based on RMSE is when using 4 variables. Since the result is purely based on RMSE, it was not considered as the best model.

Regression Model on Setting 8 is more prominent than the other seven models with smaller p-value and decent R-squared.

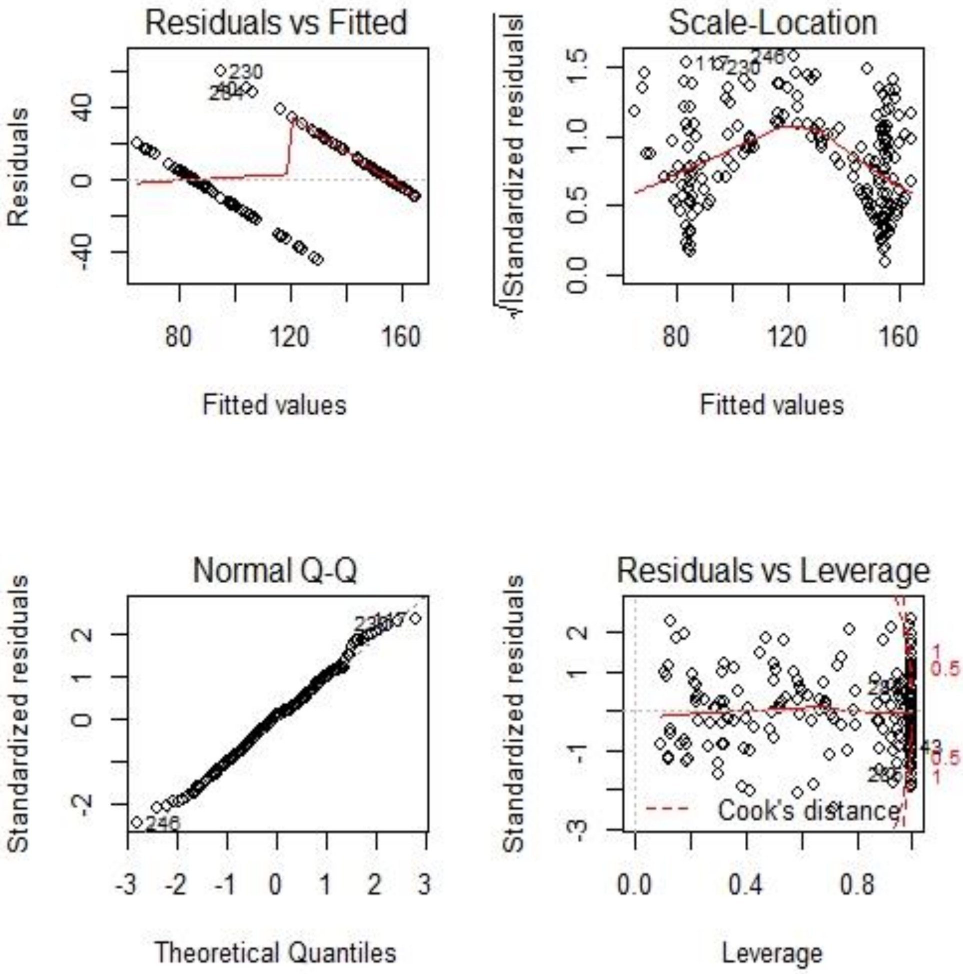


Figure 5.4: Setting 8 model plots

The model from Setting 9 can be considered as a proof of concept even though the model is complex to interpret and has the adjusted R-squared below 0.8. Model on Setting 9 showed a better result with very small p-value, elevated adjusted R-squared value, and smaller standard residual error.

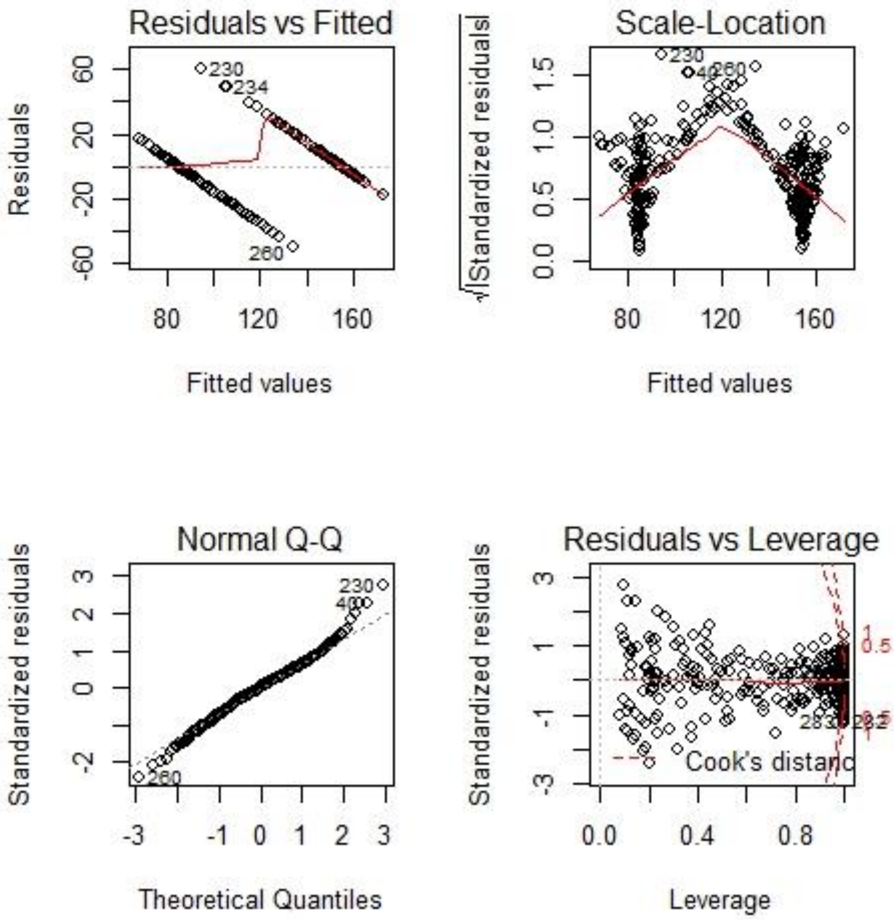


Figure 5.5: Model plots for Settings 9

Figure 5.4 and Figure 5.5 shows the four plots of the model obtained from Setting 8 and Setting 9, respectively. Since there is no parabolic pattern visible in Residual vs Fitted plots, we can assure that the model has captured the non-linear relationships between independent variables. The Normal Q-Q plot shows that the residuals are normally distributed. The scale-location plot shows that the residuals usually appear even though it is not horizontal to the x-axis; this is due to the limited number of observed values. The Residual vs Leverage graph shows that there are few rows in the dataset, which are influential observations.

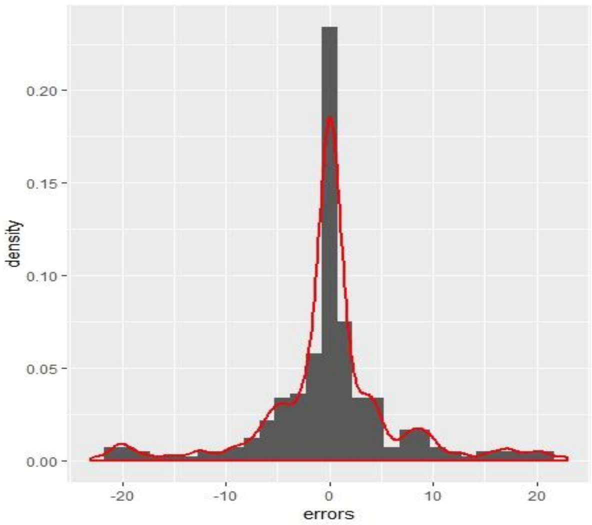


Figure 5.6: Distribution of errors for regression using settings 9

Figure 5.6 shows the error distribution of the inference testing using regression on settings 9, and it is safe to say that the regression inference predicts with of ± 21 kg for 65% of the data, which is of $\pm 19\%$ accuracy on average for 65% cases, of $\pm 38\%$ accurate with 95% confidence.

5.2 Evaluation of Bayesian Regression

Bayesian Regression was tested using all the basic features and the features in Setting 8. Unfortunately, the models did not give promising results. The best model produced a negative R-squared value (-82.43626), with a relatively large Mean Squared Error (1194.18). Figure 5.7 shows the deviation plot for the Bayesian regression prediction.

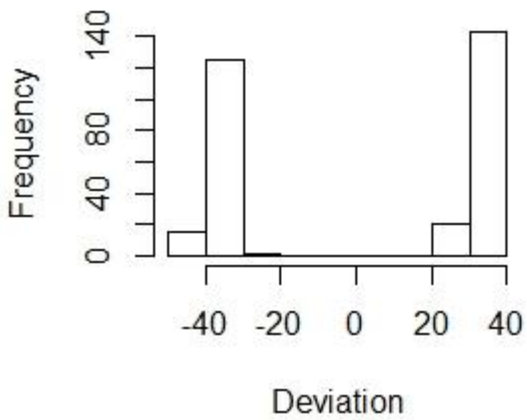


Figure 5.7: Deviation graph for bayesian regression

5.3 Evaluation of Decision Trees

Five different Regression Tree algorithms were tested using the dataset. Table 5.2 summarises the performance of the five algorithms. Random forest (Breiman, 2001) performed better than the other four algorithms. Secondly, Recursive Partitioning (Strobl, Malley and Gerhard Tutz, 2009) performed relatively better than the other three algorithms on the dataset.

Table 5.2: Summary of Regression Trees

Algorithm	R-Squared	Adjusted R-Squared	Mean Squared Error	Residual Standard Error
Recursive Partitioning	0.54	0.514719	387.21	20.15
M5P	-5.42	-5.70988	1031.69	32.88
M5Rules	-4.75	-5.00505	1028.91	32.84
Random Forest	0.67	0.656273	246.49	16.07
Cubist	-0.66	-0.73826	1021.91	32.73

The performance of Random Forest increased after having around 200 Decision Trees. Figure 5.8 shows the performance of Random Forest changes with the number of trees.

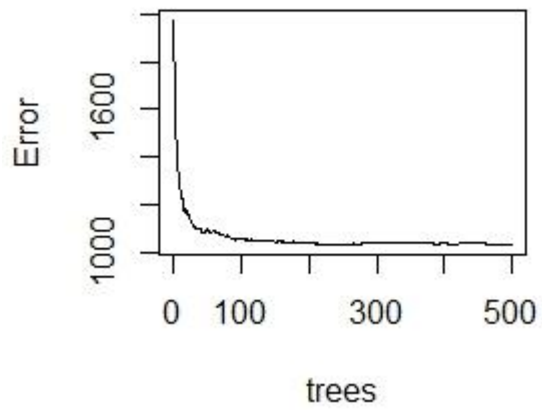
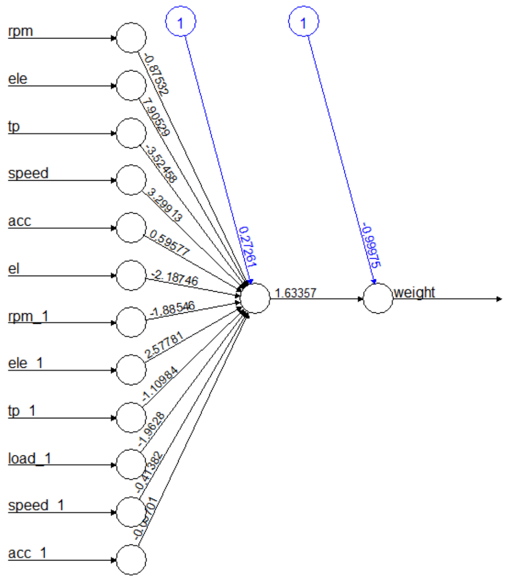


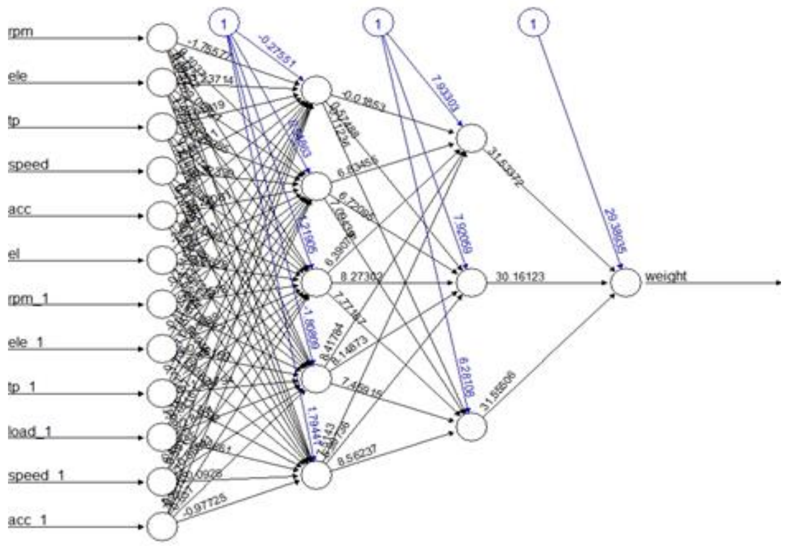
Figure 5.8: Number of trees vs error of Random Forest

5.4 Evaluation of ANN Model performance

Twenty different ANN architectures were tested with the dataset. All architectures were fed with the normalised feature values. All the 6 basic features with their inverse were used in all twenty architectures. Figure 5.9 shows two sample ANN architectures (a) with one hidden layer of one node, (b) with two hidden layers with 5 and 3 nodes in each layer, respectively.



a



b

Figure 5.9: Architecture of ANN model (a) One hidden layer with one node, (b) Two hidden layers with five, three nodes in each layer

Table 5.3 lists the twenty ANN architectures (from model 1 to model 20) and their performance. It is evident that the model performance increases with the number of nodes in hidden layers. In this research, we have stopped testing after reaching a reasonable performance, which is found in model 20.

Table 5.3: Performance of the ANN model

Model #	Hidden Layers	Nodes	Accuracy	R-Squared	Adjusted R-Squared	Mean Squared Error	Residual Std. Error
1	1	1	0.912	-18.67	-19.56	1158.72	34.85
2	1	5	0.945	-0.44	-0.51	720.57	27.48
3	1	10	0.963	0.33	0.30	485.15	22.55
4	1	20	0.984	0.80	0.79	199.48	14.46
5	1	30	0.978	0.70	0.69	276.58	17.03
6	1	40	0.986	0.84	0.83	168.64	13.30
7	1	50	0.988	0.86	0.85	149.25	12.51
8	2	5,5	0.951	-0.14	-0.19	649.19	26.08
9	2	10,5	0.970	0.53	0.51	390.99	20.24
10	2	10,10	0.980	0.72	0.71	262.78	16.60
11	2	20,10	0.992	0.91	0.90	102.27	10.35
12	2	20,20	0.992	0.91	0.90	102.27	10.35
13	2	30,20	0.995	0.95	0.95	57.29	7.75
14	2	30,30	0.996	0.96	0.96	45.39	6.90
15	3	5,5,5	0.971	0.54	0.52	386.02	20.11
16	3	10,10,10	0.990	0.88	0.87	133.02	11.81
17	3	20,20,10	0.994	0.94	0.93	72.86	8.74
18	3	30,20,10	0.996	0.96	0.96	46.81	7.00
19	3	30,30,20	0.912	0.96	0.95	50.35	7.27
20	3	30,30,30	**0.945**	**0.97**	**0.97**	**34.68**	**6.03**

Figure 5.10 contains nine selected graphs of frequency distribution of deviation for architecture 1,2,7,8,11,12,14,16, and 20. It shows the progression of ANNs performance with the number of layers and nodes. Figure 5.10 (a) – (c) have shown that the increase in the number of nodes/neurons in a layer reduces the deviation. The biggest improvement of the model performance is observed when we increase the number of clusters. The ANN architecture of Model 20 (3 hidden layers with 30 nodes in each) has shown a better performance with increased R-Squared and Adjusted R-Squared, and decreased residual standard error.

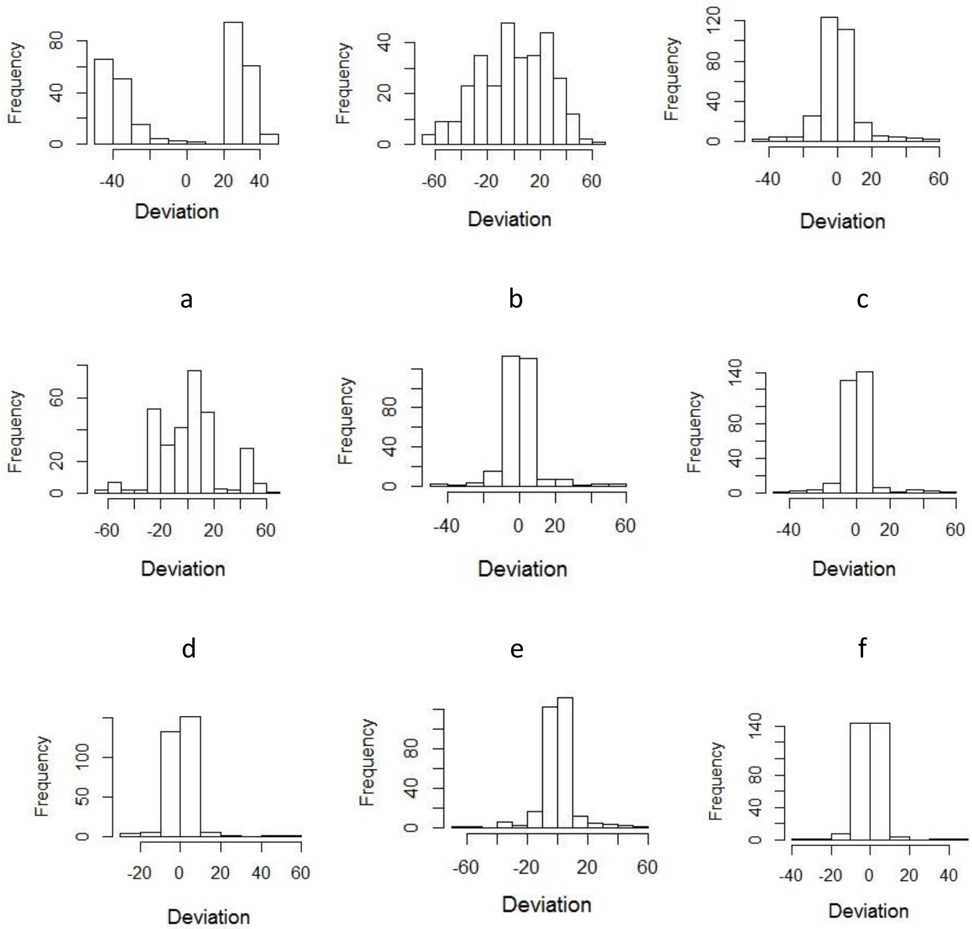


Figure 5.10: Frequency distribution of deviations showing improvement when increasing number of nodes and layers for ANN model (a) 1, (b) 2, (c) 7, (d) 8, (e) 11 , (f) 12, (g) 14, (h) 16, (i) 20

Plots in Figure 5.10 (a) – (i) shows how the deviation error shrinks from higher to lower (i.e. residual standard error form 34.85 for Model 1 to 6.03 for model 20).

5.5 ML Model performance

Table 5.4 compares the performance of each ML model on this test set. The values shown in each model are from the best runs.

Table 5.4: ML models and their performances

Model	R-Squared	Adjusted R-Squared	Residual Std. Error
Regression	0.8736	0.5633	23.1
Recursive Partitioning	0.54	0.514719	20.15
M5P	-5.42	-5.70988	32.88
M5Rules	-4.75	-5.00505	32.84
Random Forest	0.67	0.656273	16.07
ANN	0.97	0.97	6.03

These models were trained with two distinct observed dependent values with 305 observations. The model performance could be increased with more training data. However, in the real world, it would be impossible to train each vehicle with a vast dataset. Finding the optimal data points is still a researchable question. We chose two random weight data, each with a nearly equal number of observations. Multiple Linear Regression, ANN, Bayesian Regression, and Decision Tree algorithms were tested. The ANN outperformed other ML algorithms. It was observed that the model performance increased with the number of nodes in each layer. In this dissertation, we have discussed the multiple linear regression and ANN models, which have shown a better performance than other ML algorithms for the dataset.

The result of this research shows evidence of the ability to infer the vehicle weight using VT data. Results reveal that a significant level of prediction could be made using the selected features. The selected NN model has performed well (with 94.5% accuracy, R-Squared = 0.97, Adjusted R-Squared = 0.97, and residual standard error = 6.03), even on a small dataset. This is encouraging because in real-world, we cannot ask the vehicle owners to drive the vehicle several times with several different weights (i.e. size of the training data is limited in reality).

5.6 WIM System Performance

Performance of a WIM system is discussed by looking at many different factors. In here, we have compared the prototype WIM system with other WIM systems using categorical values. Table 5.5 discusses the performance comparison of the proposed WIM system approach with the existing WIN systems based on findings by Jacob and Veronique (2010), Karim *et al.* (2014), Lydon *et al.* (2016), Gajda *et al.* (2018), Timerson (2018) and the U.S. State Department (2018).

Table 5.5: Other WIM Systems Vs the Proposed System

WIM System	Type	Cost	Accuracy	Calibration Frequency	Availability	Chances of Failures	Measuring Speed
Static WIM	Stress Sensors/ Coils	High ($ 1M)	High (Restricted)	Medium	Low	Low	Stationary/ Low (10-30min)
LS-WIM	Stress Sensors	High	Moderate (Restricted)	Moderate (Annual)	Low	Moderate	Moderate
HS-WIM	Piezoelectric cable	High ($105K)	Low (Restricted)	High	Moderate	High	High
	Line quarts	High	Low (Restricted)	High	Moderate	High	High
Proposed WIM	Telematics	Low	Low (Unrestricted)	Moderate	High	Low	High

5.6.1 Cost

Cost of a WIM solution is based on installation and maintenance expenses, and labour cost. In comparison with other WIM solution, the proposed WIM system approach does not have any maintenance cost or labour cost. Additionally, the installation cost could be negligible if the existing Telematics devices are used. The main cost in this system will be maintaining the cloud server. This is way cheaper than the existing WIM systems, thus labelled low.

5.6.2 Accuracy

The accuracy of a WIM system is not homogenous throughout the entire range. WIM scale measuring the weight in several thousand kilograms (larger scale interval) may not accurately measure the smaller weights in tens of kilograms (small scale interval). The current WIMs focus on bigger vehicles such as trucks and hauling vehicles, weighing several tons. Such systems' weighting accuracy is limited to specific weight range. The range of the current WIMs excludes smaller vehicles such as cars (Haugen et al., 2016). But the proposed WIM system approach could be simply deployed on any compatible vehicles with OBDII port. The weight inference from this new proposed WIM system approach does not have any specific weighing limit (unrestricted). The static weighbridges are the most accurate in the list. But the readability (scale interval) of such Static WIMs are usually ~100kg. This is the common case for most of the WIM systems since they are used to measure the loads (weights) of heavy vehicles. This limitation in the WIM systems made us label them with restricted accuracy. The maximum reading capacity of these WIMs is up to several metric tons. But, due to the power produced by the engine is one of the features used to infer the weight, VT data from vehicles with a big engine might have poor readability, i.e. greater scale interval. This needs to be researched further.

5.6.3 Time to measure

Static WIMs are very slow in measuring, usually 10-30 minute or greater (Jacob and Véronique, 2010). Jacob and Véronique (2010) reported that the meantime between two Static WIM checks of a given truck operated every day was almost 30 years. The calibration frequency is reported higher in HS-WIMs than in Static and LS-WIMs. In the proposed system, once a vehicle is trained with VT data, the re-training can be done anytime. This re-training process can be considered as calibration in other WIMs. This can be done in case of repeated false inference.

5.6.4 Availability

Availability is the presence of WIM systems. Static WIM systems are usually located in a separate place away from the road. The LS-WIMs and HS-WIMs are placed in several road segments. But they are deployed in specific locations. The proposed WIM could be virtually available everywhere on any road segment.

5.6.5 Chance of Failure

According to literature, the sensor material used in HS-WIM is more fragile and prone to failure. Since the prototype WIM system does not use any such sensors and rely on robust ECU data, it has a lesser chance of failure.

Once the data is available on the backend server, the inference speed is nearly instantaneous. This makes the prototype system perform much faster in measuring speed. One of the most important advantages of the system is that it is scalable. The proposed WIM system approach is scalable and cost-effective, as compared to other WIM solutions. We can use the existing data collection devices used in insurance (UBI or PAYD) schemes. This would reduce the cost of implementation on a large scale. Communication technologies such as LoRaWAN offers to build fast, reliable, cheaper communication systems.

5.7 C-K theory in action

The use of C-K design theory in the design and development phase is briefly discussed here.

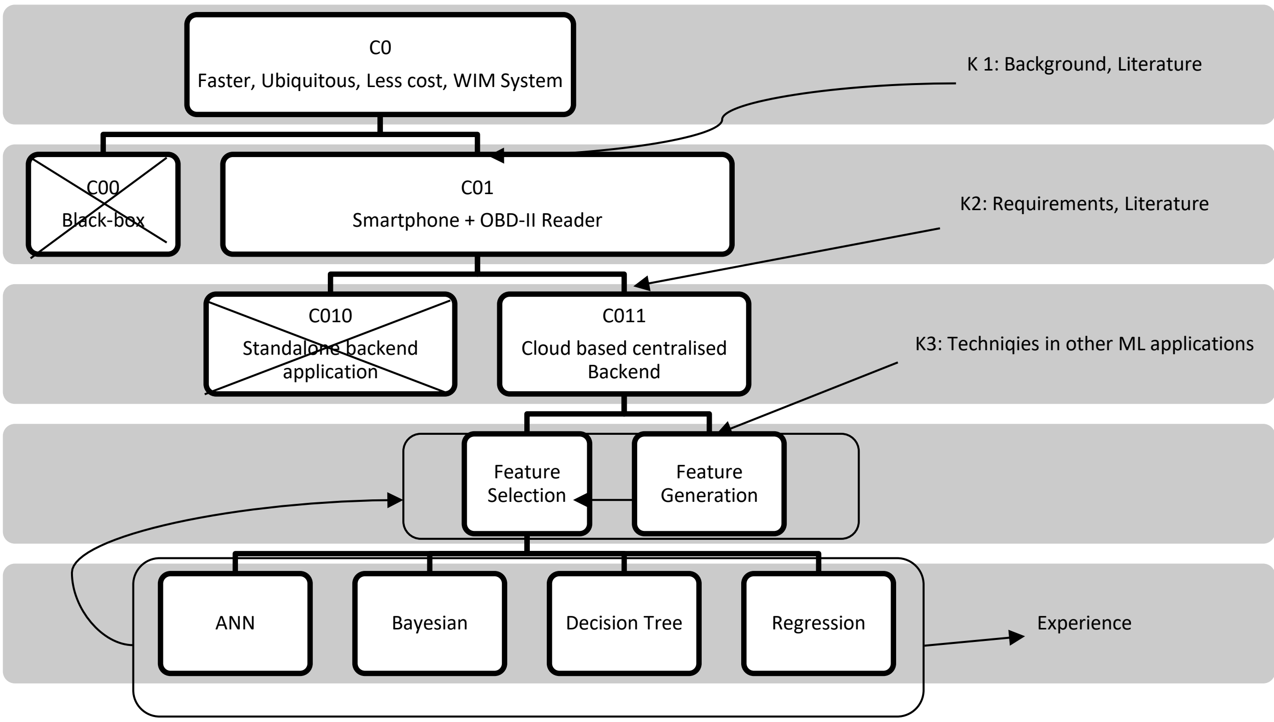


Figure 5.11: Representation of this research using typical C-K design diagram

Figure 5.11 is the simplest representation of this research using the C-K theory. The tree is built starting from the initial concept C0, by keeping the constraints (propositions) at the top level. The tree grows by listing the possible candidate solutions for the next level of implementation of the current selection.

The concept tree cannot be created at the early stage of the research. But the Tree will start growing at each stage of design and development where we make important design decisions. These stages form new levels, i.e. design spaces. In this research, we have recorded the following main stages and their substages:

Level 1. Data collection technique
- **1.1. Using a smartphone and OBD adaptor**
- 1.2. Using an existing black-box device

Level 2. Backend design
- **2.1. Using a cloud-based backend application**
  - 2.1.1. Platform
    - 2.1.1.1. Kubernetes
  - 2.1.2. Language
    - 2.1.2.1. Golang
  - 2.1.3. DBMS
    - 2.1.3.1. Cassandra
  - 2.1.4. Stream processing
    - 2.1.4.1. Akka actor framework
- 2.2. Using a simple backend application

Level 3. Feature Generation
- 3.1. Using the existing features as it is
- 3.2. Feature crossing
- 3.3. Applying various non-linear functions
- **3.4. Combining 3.2 and 3.3**

Level 4. Feature selection
- 4.1. Choose all features
- **4.2. Use feature selection techniques**

Level 5. ML algorithm
- **5.1. Neural Network**
- 5.2. Bayesian Regression
- 5.3. Decision Tree
- **5.4. Multiple Linear Regression**

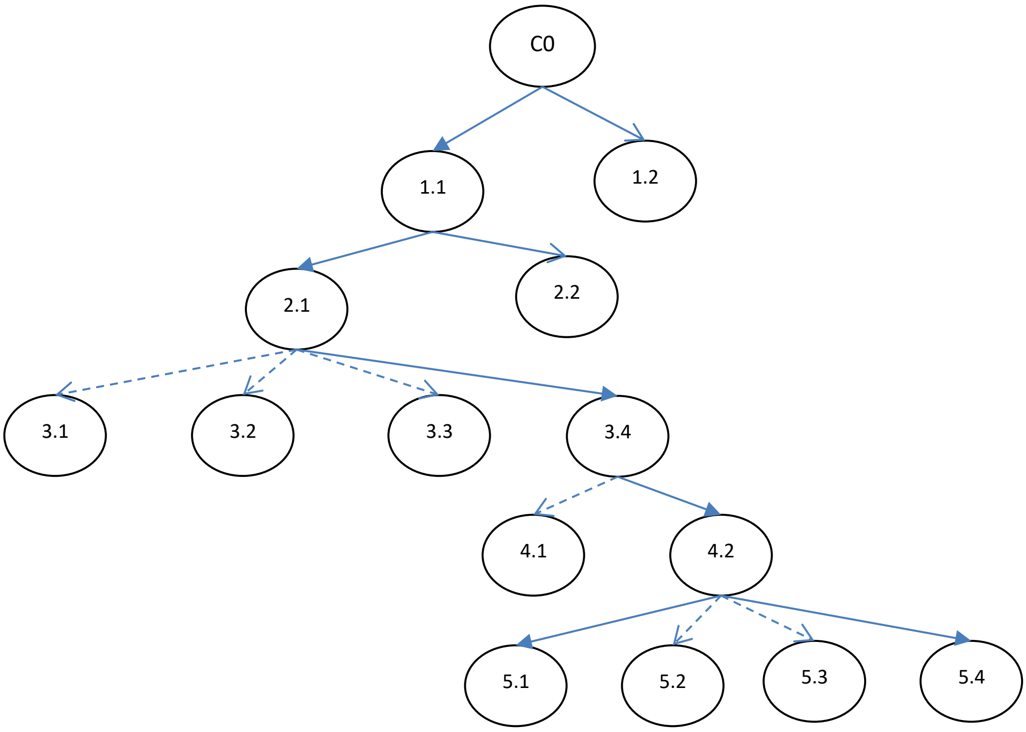


Figure 5.12: Concept Tree of new WIM system development

Figure 5.12 shows the simple Concept Tree of this research. Unlike other nodes, selection of node in Level 3 is transitively depended on level 5. The creation of new features is depended on the performance of the ML algorithm. The ML model in Level 5 is determined by the selected features in Level 4. In the meantime, Level 4 is determined by the node in Level 3. The flow from level 3 to 5 was iterative until we selected a better ML algorithm (5.4) in Level 5. Different feature engineering (Levels 3 and 4) techniques were used iteratively. Finally, we stopped iterating after reaching a decent inference accuracy (which is discussed here). Here we have used the proposed Concept Tree to discuss the decisions made during the main concept spaces. By looking at this concept tree, a researcher could understand the decisions made in each sub-concept space. This would enable a future researcher to choose a different design decision at any stage and continue the research. The adaptability of the Concept Tree in IT-based DSR, producing instantiation type of artefact still needs to be researched. There must be more DSR to be tried to report their design phase of an artefact using Concept Tree. The main drawback of this Concept tree is that this could grow bigger, and thus becomes unreadable. If there are many sub-concepts (nodes) there should be a mechanism to shrink the Concept Tree by grouping sub-concepts. It would not be able to report each sub design spaces in detail. It can only be drawn after the completion of the design. Unlike C-K theory, Concept Tree only captures the concept and sub-concepts in the design, excluding the Knowledge part of it. The next chapter concludes this dissertation by highlighting the important facts in this research.

CHAPTER 6
CONCLUSION

This dissertation concludes in the chapter by summarising the research outcomes by discussing each objective of this research, as mentioned in Chapter 1. Section 6.1 to 6.4 discuss each of the research objectives and Section 6.5 discusses the future directions.

Section 6.1. Introduction

Section 6.2. Objective1: Identify the relevant development platforms, parameters (features), and algorithms to infer the weight of a vehicle in motion.

Section 6.3. Objective 2: Design a Conceptual framework that integrates VT and ML for WIM

Section 6.4. Objective 3: Develop a prototype system that leverages VT and ML to determine the weight of a vehicle in motion.

Section 6.5. Objective 4: Evaluate the prototype system in terms of performance (accuracy, speed), usability and cost.

Section 6.6. Conclusion

Section 6.7. Future Directions

6.1 Introduction

This research was started with a systematic literature review to find a research problem. An omnipresent WIM system to monitor the payload of the vehicles 24x7 in any road segment could help the transport industry. A new WIM system approach was proposed to use VT data and ML to infer the payload of a vehicle. The concept idea was tested by developing a prototype system and evaluating in the context of small cars. The design and development of the artefact were carried out and recorded using the unified design theory called C-K theory. Concept Tree, a DSR reporting method based on C-K theory, was proposed. Adopting C-K theory in DSR was exhibited by the development of Concept Tree for the prototype design. The contribution of this research is threefold, practical, theoretical, and methodological. Artefact design and development contribute to the practical aspect of the research. The theoretical aspects of this research were covered by introducing a new concept that, under certain engines and environment conditions, the weight of a vehicle influences the values of the engine parameters. The use of C-K application of C-K design theory contributed to the methodological contribution of this research.

6.2 Identify the relevant development platforms, parameters (features), and algorithms to infer the weight of a vehicle in motion.

A systematic literature review was compiled to find the use of VT in road safety, and it is discussed in Chapter 2. While doing the review, it was found that there was some potential use of VT data to be used, one of which is the use of VT data for WIM.

The VT data collection devices and the available features collected from such VT data collection devices have been identified and reported in Chapter 2. The specially designed “black-boxes” containing embedded GNSS unit, IMU (Accelerometer/Gyroscope), and OBD adaptor were found more commonly used by some transport industries such as fleet tracking, and vehicle insurance companies. Smartphones and OBD adapters were also used in the literature of research finding driving behaviour and road anomaly detection. The quality of data from specially designed Black-boxes was reported better than using smartphone sensors. In order to save time and money, the combination of smartphones and OBD adaptors was used in this research. Features were selected from smartphone sensors and generic Standard ELM 327 OBDII data module.

6.3 Design a Conceptual framework that integrates VT and ML for WIM

Design of a Conceptual framework to consume the VT data, store the data, train and test ML model was presented in Chapter 4. The design was then developed as a simple prototype system.

6.4 Develop a prototype system that leverages VT and ML to determine the weight of a vehicle in motion

The implementation of the prototype system was discussed in Chapter 4. The Prototype used Android Smartphone and OBD-II Bluetooth adaptor as the “Black-box”. As a by-product of this research, ML backend (WIM application) was implemented and deployed as a containerised application to read and infer the weight of a vehicle. Prototype WIM application was developed using Golang. Apache Cassandra was used as the DBMS. The application was deployed on a Kubernetes cluster. The different ML algorithms were tested using the data collected. The developed system could be used to consume more VT data from different vehicles. It is possible to train and test other models with more significant data. It could also be used to build a generalized inference model. Currently, the prototype was tested using the data form a single gasoline vehicle.
The application of C-K design theory in this DSR was showcased by introducing a simple Concept Tree diagram.

6.5 Evaluate the prototype system in terms of performance (accuracy, speed), usability and cost.

Chapter 5 discussed the performance of algorithms which yielded better results. Significantly multiple polynomial regression has performed relatively better with the data collected during the research. The comparison of the prototype system with the existing system was discussed with the help of a table in Chapter 5.

6.5.1 Assumptions and Limitations

This research was based on the following assumptions and limitations:

Tyre

According to Mckay *et al.* (2012), tyre pressure influences the detection accuracy. The recommended tyre pressure was maintained, and the pressure fluctuation due to the atmospheric temperature change was neglected. Influence of the size and the shape of tyres (tyre profile) was not considered in this research.

Influence of Weather

This research was done ignoring external weather factors such as extreme wind, snow, and rain. The datasets used in this research only contains data collected during calm sunny days.

Road conditions

Friction quotient is a significant factor for moving a vehicle without slipping. Road conditions and types of roads play the primary role in friction. This factor was not considered in this research as all the data were collected from urban carpet roads.

Shifting Pattern

The gear shifting pattern and clutch releasing pattern may differ from person to person. This could influence the transmission function on manual transmission vehicles. This model was built using a single driver driving data.

Boosted and Hybrid power

Turbocharged and hybrid vehicles may produce different results as the EL formula does not apply to those vehicles. This research did not focus on such types of vehicles.

6.6 Conclusion

This research has three contributions, theoretical, practical, and methodological. Theorising that *"the engine control unit data, speed, and the road condition of a vehicle are related to the gross weight of the vehicle"* contributes to the theoretical aspect of this research. The implementation of the prototype WIM system contributes to the practical aspect of this research. Application of C-K design theory in this research, and the introduction of Concept Tree for recording DSR prototype development using C-K theory contribute to the methodological contribution of this research. A prototype WIM system, as proposed, was developed and tested. The pragmatic validity of a generic design refers to the question of whether it will work after contextualisation and implementation (Aken, Chandrasekaran and Halman, 2016). This research could be reproduced in different contexts (types of vehicles) to generalise the results inductively. Several possible feature variables and ML algorithms were tested to build a better ML model. It is found that the multiple non-linear regression model from Settings 9 performs better than linear and non-linear Regression models, with the smaller Residuals Standard Error = 23.1, degrees of freedom = 88, higher significance p-value = 6.322e-08, better R-squared = 0.87, and a decent adjusted R-squared = 0.56.

In the meantime, an ANN architecture of three hidden layers with 30 nodes in each layer has shown astounding performance with Accuracy = 0.945, R-Squared = 0.97, Adjusted R-Squared = 0.97, Mean Squared Error = 34.68, Residual Standard Error = 6.03. The ANN outperformed all other tested ML algorithms on the collected VT dataset. The size of the training dataset is crucial in this research as it will not be feasible to obtain a large dataset for vehicles in real-world scenarios. In this research, we have obtained an ML model with decent inference accuracy using a smaller dataset. We found that the VT data from 0 to 20kmph, particularly during the first gear produces more prominent result than the entire VT data.

Model performance result shows that, in context of a small car, it is possible to infer the payload using the instantaneous VT data such as RPM, Road Gradient (elevation), Vehicle Speed, Acceleration, and Calculated Engine load. The inference accuracy of the prototype WIM system was ±7kg in average for 65% of the data, which is ± 6% accuracy in average for 65% cases, and of ±12% accurate with a 95% confidence level. Even though the accuracy is not up to the level of existing Static WIM systems, the performance of the prototype on the collected data shows better accuracy compared to the standard HS-WIMs. This preliminary result opens future research space. Longitudinal research considering weather factors, tyre profile, road conditions, and other engine types may produce a significant result in this research area. This research has shown the possibility of using VT data to infer the vehicle weight. This could be adopted by the transport industry to perform shallow screening to filter possible overloaded vehicles, in any road segment at any time. The comparison of the prototype of the proposed WIM system approach with the existing systems shows that the proposed WIM system is a cheaper, scalable, omnipresent, and online (24/7) solution.

6.7 Future Directions

The optimal training size of the dataset for a better model performance still needs to be researched. Further empirical research is needed, considering tyre pressure, weather, and different shifting patterns. In addition to that, the following are some of the future research directions:

- Detecting the accuracy of different engine sizes:

Here we used a small capacity engine vehicle to test our assumptions. The smallest unit of 30kg was able to infer by using the vehicular test data. It is due to the torque or the engine capacity of the vehicle. Finding the minimum unit of mass, which can be inferred from other types of vehicles needs to be researched. This would help us to find how the capacity of the machine is influenced in determining the mass of the vehicle.

- Building a generalised inference model based on several vehicle types:

Using VT data from several other vehicles could help us to build a generalised weight inference model for most of the vehicles.

REFERENCES

Adell, E. *et al.* (2008) 'Developing human–machine interaction components for a driver assistance system for safe speed and safe distance', *IET Intelligent Transport Systems*, 2(1), p. 1. doi: 10.1049/iet-its:20070009.

Aken, J. Van, Chandrasekaran, A. and Halman, J. (2016) 'Conducting and publishing design science research: Inaugural essay of the design science department of the Journal of Operations Management', *Journal of Operations Management*. Elsevier B.V., 47–48, pp. 1–8. doi: 10.1016/j.jom.2016.06.004.

Akshay Kumar H and Suresh, Y. (2016) 'Multilayer feed forward neural network to predict the speed of wind', in *2016 International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS)*. IEEE, pp. 285–290. doi: 10.1109/CSITSS.2016.7779372.

Al-Qadi, I. *et al.* (2016) *LTBP program's literature review on weigh-in-motion systems*, *United States Department of Transportation, Federal Highway Administration*.

Alghamdi, A. H. (2013) 'Adapting Design-Based Research as a Research Methodology in Educational Settings', *International Journal of Education and Research*, 1(10), pp. 1–12. doi: 10.1016/j.emj.2010.10.004.

Amarasinghe, M. *et al.* (2015) 'Cloud-Based Driver Monitoring and Vehicle Diagnostic with OBD2 Telematics', in *IEEE International Conference on Electro/Information Technology (EIT)*. IEEE, pp. 505–510. doi: 10.1109/EIT.2015.7293433.

Anthony, P. C. (2013) 'Method for detecting gross vehicle weight overload'. Bae Systems Information And Electronic Systems Integration Inc.

Armand Hatchuel and B Weil (2003) 'A New Approach of Innovative Design: an Introduction To C-K theory', in *International Conference on Engineering Design*, pp. 1–15. doi: citeulike-article-id:4891368.

Asif, F. *et al.* (2009) 'Haptic controls in cars for making driving more safe', in *2009 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, pp. 2023–2028. doi: 10.1109/ROBIO.2009.5420520.

Bajwa, R. *et al.* (2013) 'In-pavement wireless weigh-in-motion', *Proceedings of the 12th international conference on Information processing in sensor networks - IPSN '13*, p. 103. doi: 10.1145/2461381.2461397.

Barsanescu, P., Carlescu, P. and Stefanescu, D. M. (2007) 'A new weigh-in-motion and traffic monitoring system', in *IMEKO 20th TC3, 3rd TC16 and 1st TC22 International Conference Cultivating metrological knowledge*. Merida, Mexico., pp. 2–7.

Belz, S. M., Robinson, G. S. and Casali, J. G. (1999) 'A New Class of Auditory Warning Signals for Complex Systems: Auditory Icons', *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 41(4), pp. 608–618. doi: 10.1518/001872099779656734.

Bircanoglu, C. and Arica, N. (2018) 'A comparison of activation functions in artificial neural networks', in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, pp. 1–4. doi: 10.1109/SIU.2018.8404724.

Braun, D., Reiter, E. and Siddharthan, A. (2015) 'Creating Textual Driver Feedback from Telemetric Data', in *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*. Brighton: Association for Computational Linguistics, pp. 156–165.

Breiman, L. (2001) *Random Forests. Otras caracteristicas*. doi: 10.1017/CBO9781107415324.004.

Bushman, R. and Pratt, A. J. (1998) ‘Weigh In Motion Technology - Economics and Performance’, in *NATMEC 98 : North American Travel Monitoring Exhibition and Conference*.

Camarinha-matos, L. M. (2012) *SCIENTIFIC RESEARCH Unit 1 : INTRODUCTION Some key questions*.

Castignani, G. *et al.* (2015a) ‘Driver behavior profiling using smartphones: A low-cost platform for driver monitoring’, *IEEE Intelligent Transportation Systems Magazine*, 7(1), pp. 91–102. doi: 10.1109/MITS.2014.2328673.

Castignani, G. *et al.* (2015b) ‘Validation study of risky event classification using driving pattern factors’, *2015 IEEE Symposium on Communications and Vehicular Technology in the Benelux, SCVT 2015*. doi: 10.1109/SCVT.2015.7374228.

Castignani, G., Frank, R. and Engel, T. (2013) ‘Driver behavior profiling using smartphones’, *16th International IEEE Conference on Intelligent Transportation Systems (ITSC 2013)*, (Itsc), pp. 552–557. doi: 10.1109/ITSC.2013.6728289.

Chan, A. B. and Vasconcelos, N. (2012) ‘Counting People With Low-Level Features and Bayesian Regression’, *IEEE Transactions on Image Processing*, 21(4), pp. 2160–2177. doi: 10.1109/TIP.2011.2172800.

Chaovalit, P., Saiprasert, C. and Pholprasit, T. (2013) ‘A method for driving event detection using SAX on smartphone sensors’, in *2013 13th International Conference on ITS Telecommunications (ITST)*. IEEE, pp. 450–455. doi: 10.1109/ITST.2013.6685587.

Chen, D. *et al.* (2015) ‘Invisible Sensing of Vehicle Steering with Smartphones’, *MobiSys*. doi: 10.1145/2742647.2742659.

Chen, Z. *et al.* (2015) ‘D3: Abnormal driving behaviors detection and identification using smartphone sensors’, in *2015 12th Annual IEEE International Conference on Sensing, Communication, and Networking, SECON 2015*, pp. 524–532. doi: 10.1109/SAHCN.2015.7338354.

CSIR (Roads and Transport Technology) (1997) *THE DAMAGING EFFECTS OF OVERLOADED HEAVY*. PRETORIA, Republic of South Africa. doi: ISBN: 1-86844-285-3.

CSS, E. (2018) *OBD2 Explained - A Simple Intro (2019)*, *CSS Electronics*. Available at: https://www.csselectronics.com/screen/page/simple-intro-obd2-explained (Accessed: 16 January 2019).

Cummings, M. *et al.* (2007) ‘Effects of single versus multiple warnings on driver performance’, *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 49(6), p. 1097. doi: 10.1518/001872007X249956.

CWCSA (2017) *Ways of preventing vehicle overloading: The effects of overloading a vehicle*, *CWCSA*. Available at: http://www.cwcsa.com/wp-content/uploads/Ways_of_preventing_a_vehicle_becoming_overloaded.pdf (Accessed: 12 December 2017).

Dai, J. *et al.* (2010) ‘Mobile phone based drunk driving detection’, *Pervasive Computing Technologies for Healthcare PervasiveHealth 2010 4th International Conference onNO PERMISSIONS*, pp. 1–8. doi: 10.4108/ICST.PERVASIVEHEALTH2010.8901.

Dogan, E. *et al.* (2011) ‘The effects of non-evaluative feedback on drivers’ self-evaluation and performance’, *Accident Analysis and Prevention*, 45, pp. 522–528. doi: 10.1016/j.aap.2011.09.004.

Domingos, P. (2012) ‘A few useful things to know about machine learning’, *Communications of the ACM*, 55(10), p. 78. doi: 10.1145/2347736.2347755.

Donald E. Stokes (1997) *Pasteur’s Quadrant*. Brookings Institution Press.

Donmez, B., Boyle, L. N. and Lee, J. D. (2007) ‘Safety implications of providing real-time feedback to distracted drivers’, *Accident Analysis and Prevention*, 39, pp. 581–590. doi: 10.1016/j.aap.2006.10.003.

Elmelectronics (2011) *OBD to RS232 Interpreter ELM327*, *Electronics*.

Engelbrecht, J. *et al.* (2015) ‘Performance Comparison of Dynamic Time Warping (DTW) and a Maximum Likelihood (ML) Classifier in Measuring Driver Behavior with Smartphones’, in *2015 IEEE Symposium Series on Computational Intelligence*. IEEE, pp. 427–433. doi: 10.1109/SSCI.2015.70.

Engelbrecht, Jarret *et al.* (2015) ‘Survey of smartphone-based sensing in vehicles for intelligent transportation system applications’, *IET Intelligent Transport Systems*, 9(10), pp. 924–935. doi: 10.1049/iet-its.2014.0248.

Farmer, C. M., Kirley, B. B. and McCartt, A. T. (2010) ‘Effects of in-vehicle monitoring on the driving behavior of teenagers’, *Journal of Safety Research*, 41(1), pp. 39–45. doi: 10.1016/j.jsr.2009.12.002.

Fazeen, M and Gozick, B and Dantu, R and Bhukhiya, M. and G. M. . *et al.* (2012) ‘Safe driving using mobile phones’, *IEEE Transactions on Intelligent Transportation Systems*, 13(3), pp. 1462–1468. doi: 10.1109/TITS.2012.2187640.

Ferreira, J. *et al.* (2017) ‘Driver behavior profiling: An investigation with different smartphone sensors and machine learning’, *Plos One*. Public Library of Science, 12(4), p. e0174959. doi: 10.1371/journal.pone.0174959.

Fleming, B. (2010) ‘Global Positioning System and Telematics [Automotive Electronics’, *IEEE Vehicular Technology Magazine*, June, pp. 4–7. doi: 10.1109/MVT.2010.936652.

Frank, E. *et al.* (2000) ‘Technical Note: Naive Bayes for Regression’, *Machine Learning*. Kluwer Academic Publishers, 41(1), pp. 5–25. doi: 10.1023/A:1007670802811.

Gajda, J., Burnos, P. and Sroka, R. (2018) ‘Accuracy Assessment of Weigh-in-Motion Systems for Vehicle’s Direct Enforcement’, *IEEE Intelligent Transportation Systems Magazine*, pp. 88–94. doi: 10.1109/MITS.2017.2776111.

Gluster, T. (2019) *Architecture - Gluster Docs*, *Red Hat*. Available at: https://docs.gluster.org/en/latest/Quick-Start-Guide/Architecture/ (Accessed: 24 May 2019).

Haik, Y. and Shahin, T. (2014) *Engineering Design Process Second Edition Yousef Haik Tamer Shahin*.

Hansheng Lei and Govindaraju, V. (2004) ‘Regression time warping for similarity measure of sequence’, *The Fourth International Conference onComputer and Information Technology, 2004. CIT ’04.* IEEE, pp. 826–830. doi: 10.1109/cit.2004.1357297.

Hatchuel, A. *et al.* (2004) ‘C-K Theory in Practice : Lessons From Industrial Applications’, *Design Conference 2004*, pp. 245–258.

Hatchuel, A. *et al.* (2017) *C-K THEORY : MODELLING CREATIVE THINKING AND ITS IMPACT ON RESEARCH*.

Hatchuel, A., Masson, P. le and Weil, B. (2006) ‘THE DESIGN OF SCIENCE BASED PRODUCTS: AN INTERPRETATION AND MODELLING WITH C-K THEORY’, in *INTERNATIONAL DESIGN CONFERENCE - DESIGN 2006*.

Hatchuel, A. and Weil, B. (2002) ‘C-K theory: Notions and applications of a unified design theory’, in *Proceedings of the Herbert Simon International Conference on « Design Sciences »*, pp. 1–22.

Hatchuel, A. and Weil, B. (2007) ‘Design as forcing: Deepening the foundations of C-K theory’, in *16th International Conferenece on Engineering Design*, pp. 1–12.

Hatchuel, A. and Weil, B. (2009) ‘C-K design theory: An advanced formulation’, *Research in Engineering Design*, 19(4), pp. 181–192. doi: 10.1007/s00163-008-0043-4.

Hatchuel, Masson and Weil (2004) ‘C-K Theory in Practice : Lessons From Industrial Applications’, *Design Conference 2004*, pp. 245–258.

Haugen, T. *et al.* (2016) ‘Weigh-in-Motion equipment – experiences and challenges’, *Transportation Research Procedia*. Elsevier B.V., 14(2352), pp. 1423–1432. doi: 10.1016/j.trpro.2016.05.215.

Hevner *et al.* (2004) ‘Design Science in Information Systems Research’, *MIS Quarterly*. doi: 10.2307/25148625.

Hevner *et al.* (2010) ‘Design Science Research Frameworks’, in *Design Research in Information Systems Theory and Practice*, pp. 9–22. doi: 10.1007/978-1-4419-5653-8_2.

Hevner, A. and Chatterjee, S. (2010) ‘Design Research in Information Systems’, in *Design Research in Information Systems,IntegratedSeries in Information Systems 22*, pp. 9–22. doi: 10.1007/978-1-4419-5653-8.

Hevner, A. R. (2007) ‘A Three Cycle View of Design Science Research’, *Scandinavian Journal of Information Systems*, 19(2), pp. 87–92. doi: http://aisel.aisnet.org/sjis/vol19/iss2/4.

Hiraman, B. R., Viresh, M. C. and Abhijeet, C. K. (2018) ‘A Study of Apache Kafka in Big Data Stream Processing’, in *2018 International Conference on Information, Communication, Engineering and Technology, ICICET 2018*. doi: 10.1109/ICICET.2018.8533771.

Hocking, R. R. (1976) ‘A Biometrics Invited Paper. The Analysis and Selection of Variables in Linear Regression’, *Biometrics*. International Biometric Society, 32(1), p. 1. doi: 10.2307/2529336.

Hong, J.-H., Margines, B. and Dey, A. K. (2014) ‘A smartphone-based sensing platform to model aggressive driving behaviors’, in *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI ’14*, pp. 4047–4056. doi: 10.1145/2556288.2557321.

Hosseinioun, S. V., Al-Osman, H. and Saddik, A. El (2016) ‘Employing Sensors and Services Fusion to Detect and Assess Driving Events’, *Proceedings - 2015 IEEE International Symposium on Multimedia, ISM 2015*. IEEE, pp. 395–398. doi: 10.1109/ISM.2015.121.

Huang, C., Zhang, J. and Yi, Z. (2009) ‘Research on Vehicle Overloading Influenced to Roadbed’, in *Logistics*, pp. 2528–2533. doi: 10.1061/40996(330)373.

Huang, Y.-H. *et al.* (2005) ‘Feedback by technology: Attitudes and opinions of truck drivers’, *Transportation Research Part F*, 8, pp. 277–297. doi: 10.1016/j.trf.2005.04.005.

Hurwitz, J. and Kirsch, D. (2018) *Machine Learning for Dummies IBM Limited Edition*, *John Wiley & Sons, Inc.* John Wiley & Sons, Inc. doi: 10.1016/B978-0-12-279671-5.50019-8.

Ihiguro, T. *et al.* (2006) ‘Vehicle Weight Estimating Device’. Japan: United States Patent.

International Organization for Standardization (2015) *ISO 15031 - Road vehicles -- Communication between vehicle and external equipment for emissions-related diagnostics*, *ISO*.

J.Labarrere (2017) *What is WIM and History*, *ISWIM - International Society for Weigh in Motion*. Available at: http://www.is-wim.org/index.php?nm=2&nsm=1&lg=en (Accessed: 14 December 2017).

Jacob, B. and Véronique, F. L. B. (2010) ‘Improving truck safety: Potential of weigh-in-motion technology’, *IATSS Research*. Elsevier Ltd, 34(1), pp. 9–15. doi: 10.1016/j.iatssr.2010.06.003.

Johnson, D. A. and Trivedi, M. M. (2011) 'Driving style recognition using a smartphone as a sensor platform', in *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, pp. 1609–1615. doi: 10.1109/ITSC.2011.6083078.

Jonas Bonér *et al.* (2014) *The Reactive Manifesto*. Available at: https://www.reactivemanifesto.org/ (Accessed: 20 April 2019).

Jonck, J. (2017) *Overloading and Road Safety*, *www.arrivealive.co.za*. Available at: https://www.arrivealive.co.za/Overloading-and-Road-Safety (Accessed: 12 December 2017).

Kalra, N., Chugh, G. and Bansal, D. (2014) 'Analyzing Driving and Road Events via Smartphone', *International Journal of Computer Applications*, 98(12), pp. 975–8887.

Karim, M. R. *et al.* (2014) 'Effectiveness of vehicle weight enforcement in a developing country using weigh-in-motion sorting system considering vehicle by-pass and enforcement capability', *IATSS Research*, 37(2), pp. 124–129. doi: 10.1016/j.iatssr.2013.06.004.

Kassambara, A. (2018) *Stepwise Regression Essentials in R - Articles - STHDA*, *STHDA*. Available at: http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/154-stepwise-regression-essentials-in-r/ (Accessed: 10 February 2019).

Kazakci, A. A., Hatchuel, A. and Weil, B. (2008) 'A model of CK design theory based on term logic: a formal CK background for a class of design assistants', *Proceedings of the DESIGN 2008, 10th International Design Conference, Dubrovnik, Croatia*, (containing M), pp. 43–52.

Khandani, S. (2005) *Engineering Design Process : Education Transfer Plan*.

Khedkar, S. and Ravi, S. (2015) 'Driver Evaluation System Using Mobile Phone and OBD-II System', 6(3), pp. 2738–2745.

Kiencke, U., Dais, S. and Litschel, M. (1986) 'Automotive Serial Controller Area Network', in *SAE International Congress and Exposition* . SAE International, p. 8. doi: 10.4271/860391.

Kim, S. *et al.* (2009) 'Vehicle Signal Analysis Using Artificial Neural Networks for a Bridge Weigh-in-Motion System', *Sensors*, 9(10), pp. 7943–7956. doi: 10.3390/s91007943.

Kirushanth, S. and Kabaso, B. (2018) 'Telematics and Road Safety', in *2nd International Conference on Telematics and Future Generation Networks*. IEEE, pp. 103–108. doi: 10.1109/TAFGEN.2018.8580482.

Kirushanth, S. and Kabaso, B. (2020) 'Design and Development of Weigh-In-Motion Using Vehicular Telematics', *Journal of Sensors*, 2020, pp. 1–22. doi: 10.1155/2020/7871215.

Kirushnath, S. and Kabaso, B. (2018) 'Weigh-in-motion Using Machine Learning and Telematics', in *2nd International Conference on Telematics and Future Generation Networks*. IEEE, pp. 115–120. doi: 10.1109/TAFGEN.2018.8580463.

Kruger, J.-L., Hefer, E. and Matthew, G. (2013) 'Measuring the Impact of Subtitles on Cognitive Load: Eye Tracking and Dynamic Audiovisual Texts', in *Proceedings of Eye Tracking South Africa Cape Town*, pp. 29–31.

Lan, J. *et al.* (2011) 'Vehicle detection and classification by measuring and processing magnetic signal', *Measurement: Journal of the International Measurement Confederation*, 44(1), pp. 174–180. doi: 10.1016/j.measurement.2010.09.044.

Lee, J. D. (2007) 'Technology and teen drivers', *Safety Research*, 38, pp. 203–213. doi: 10.1016/j.jsr.2007.02.008.

Lin, S.-C. and Li, K.-C. (2016) 'Systems and methods for vehicle load detection and response'. US.

Liu, Z. *et al.* (2016) 'SenSafe: A Smartphone-Based Traffic Safety Framework by Sensing Vehicle and Pedestrian Behaviors', *Mobile Information Systems*. Hindawi, 2016, pp. 1–13.

doi: 10.1155/2016/7967249.

LoRa Alliance (2015) *LoRaWAN: What it is?*

LoRa Alliance (2019) *About LoRaWAN® | LoRa Alliance™*. Available at: https://lora-alliance.org/about-lorawan (Accessed: 9 April 2019).

Lu, S. C.-Y. and Liu, A. (2012) ‘Abductive reasoning for design synthesis’, *CIRP Annals*. Elsevier, 61(1), pp. 143–146. doi: 10.1016/J.CIRP.2012.03.062.

Lydon, M. *et al.* (2016) ‘Recent developments in bridge weigh in motion (B-WIM)’, *Journal of Civil Structural Health Monitoring*. Springer Berlin Heidelberg, 6(1), pp. 69–81. doi: 10.1007/s13349-015-0119-6.

Magaña, V. C. and Muñoz-Organero, M. (2017) ‘Toward Safer Highways’, *IEEE Vehicular Technology Magazine*, pp. 69–76. doi: 10.1109/MVT.2017.2692059.

Malhotra, R. (2015) ‘A systematic review of machine learning techniques for software fault prediction’, *Applied Soft Computing*, 27, pp. 504–518. doi: 10.1016/j.asoc.2014.11.023.

March, S. T. and Gerald, F. S. (1995) ‘Design and natural science research on information technology Salvatore’, *Decision Support Systems*, 15, pp. 251–266.

March, S. T. and Smith, G. F. (1995) ‘Design and natural science research on information technology’, *Decision Support Systems*, 15(4), pp. 251–266. doi: 10.1016/0167-9236(94)00041-2.

Massaro, E. *et al.* (2016) ‘The Car as an Ambient Sensing Platform [Point of View]’, *Proceedings of the IEEE*, 105(1), pp. 3–7. doi: 10.1109/jproc.2016.2634938.

Matsuzaki, R. and Todoroki, A. (2008) ‘Wireless monitoring of automobile tires for intelligent tires’, *Sensors*, 8(12), pp. 8123–8138. doi: 10.3390/s8128123.

McCarthy, J. (1980) ‘Circumscription—A form of non-monotonic reasoning’, *Artificial Intelligence*. Elsevier, 13(1–2), pp. 27–39. doi: 10.1016/0004-3702(80)90011-9.

Mcgehee, D. V *et al.* (2007) ‘Extending parental mentoring using an event-triggered video intervention in rural teen drivers’, *Journal of Safety Research*, 38, pp. 215–227. doi: 10.1016/j.jsr.2007.02.009.

Mckay, T. R. *et al.* (2012) ‘Passive detection of vehicle loading’, in *SPIE-IS&T Electronic Imaging*. doi: 10.1117/12.912100.

Meiring, G. A. M. and Myburgh, H. C. (2015) ‘A review of intelligent driving style analysis systems and related artificial intelligence algorithms’, *Sensors (Switzerland)*, 15(12). doi: 10.3390/s151229822.

Meseguer, J. E. *et al.* (2013) ‘DrivingStyles: A smartphone application to assess driver behavior’, in. IEEE, pp. 000535–000540. doi: 10.1109/ISCC.2013.6755001.

Mina Andrawos and Martin Helmich (2017) *Cloud Native Programming with Golang*. Packt.

Mishra, K. (2018) *India Automated Illegal Parking Management LoRaWAN ™ Whitepaper*.

Moosavi, S., Ramnath, R. and Nandi, A. (2016) ‘Discovery of driving patterns by trajectory segmentation’, in *Proceedings of the 3rd ACM SIGSPATIAL PhD Symposium on - SIGSPATIAL PhD ’16*. New York, New York, USA: ACM Press, pp. 1–4. doi: 10.1145/3003819.3003824.

Mwilu, S. *et al.* (2016) ‘Design science research contribution to business intelligence in the cloud — A systematic literature review’, *Future Generation Computer Systems*. Elsevier B.V., 63(2016), pp. 108–122. doi: 10.1016/j.future.2015.11.014.

Nichols, A. P. and Bullock, D. M. (2004) ‘Quality Control Procedures for Weigh-in-Motion

Data', *Transportation Research*, p. 255. doi: 10.5703/1288284313299.

Nunamaker, J. F., Chen, M. and Purdin, T. D. M. (1990) 'Systems Development in Information Systems Research', *Journal of Management Information Systems*. Routledge, 7(3), pp. 89–106. doi: 10.1080/07421222.1990.11517898.

Oastler, M. (2015) *Vehicle weights explained | tare, kerb, GVM, payload and trailer figures - Car Advice | CarsGuide*. Available at: https://www.carsguide.com.au/car-advice/vehicle-weights-explained-tare-kerb-gvm-payload-and-trailer-figures-37482 (Accessed: 21 December 2017).

Offermann, P. *et al.* (2009) 'Outline of a design science research process', p. 1. doi: 10.1145/1555619.1555629.

Oh, D. H. (2018) *Cheat Sheet : Kubernetes*.

Ouyang, Z. *et al.* (2016) 'Multiwave: A novel vehicle steering pattern detection method based on smartphones', in *2016 IEEE International Conference on Communications (ICC)*. IEEE, pp. 1–7. doi: 10.1109/ICC.2016.7511088.

Pais, J. C., Amorim, S. I. R. and Minhoto, M. J. C. (2013) 'Impact of traffic overload on road pavement performance', *Journal of Transportation Engineering*, 139(9). doi: 10.1061/(ASCE)TE.1943-5436.0000571.

Pargaonkar, S. *et al.* (2014) 'Safe Driving Using Android Based Device', *International Journal of Engineering Trends and Technology (IJETT)*, 18, pp. 151–154.

Paul Hilgeman, S. K. W. and Vicente, R. (2000) 'SYSTEM FOR CONTROLLING ACCELERATION OF A LOAD COUPLED TO A GAS TURBINE ENGINE', *Time*. US: United States Patent.

Peffers, K. *et al.* (2008) 'A Design Science Research Methodology for Information Systems Research', *Journal of Management Information Systems*, 24(3), pp. 45–77. doi: 10.2753/mis0742-1222240302.

Pirelli (2017) *Pirelli Connesso, the smart tyre that interacts with the driver*, *Pirelli & C. S.p.A*. Available at: https://www.pirelli.com/global/en-ww/road/pirelli-connesso-the-smart-tyre-that-interacts-with-the-driver# (Accessed: 12 December 2017).

Pohlmann, J. T. and Leitner, D. W. (2003) 'A Comparison of Ordinary Least Squares and Logistic Regression', *The Ohio Journal of Science*.

Ranjit, K. (2011) *Research Methodology*. Third. SAGE Publications Ltd.

Razavi, S. and Tolson, B. A. (2011) 'A New Formulation for Feedforward Neural Networks', *IEEE Transactions on Neural Networks*, 22(10), pp. 1588–1598. doi: 10.1109/TNN.2011.2163169.

Richardson, J. *et al.* (2014) 'On the use of bridge weigh-in-motion for overweight truck enforcement', *International Journal of Heavy Vehicle Systems*, 21(2), p. 83. doi: 10.1504/IJHVS.2014.061632.

RoSPA (2013) *Road Safety and In-vehicle Monitoring (Black Box) Technology Policy Paper*.

Saiprasert, C., Pholprasit, T. and Thajchayapong, S. (2017) 'Detection of Driving Events using Sensory Data on Smartphone', *International Journal of Intelligent Transportation Systems Research*. Springer US, 15(1), pp. 17–28. doi: 10.1007/s13177-015-0116-5.

Santa, J. *et al.* (2019) 'LPWAN-Based Vehicular Monitoring Platform with a Generic IP Network Interface', *Sensors*. Multidisciplinary Digital Publishing Institute, 19(2), p. 264. doi: 10.3390/s19020264.

Schickore and Jutta (2014) *Scientific Discovery, The Stanford Encyclopedia of Philosophy*.

Spring 201. Edited by E. N. Zalta. Stanford University.

Selvaganesan, M. and Liazudeen, M. A. (2016) ‘An Insight about GlusterFS and Its Enforcement Techniques’, in *2016 International Conference on Cloud Computing Research and Innovations (ICCCRI)*. IEEE, pp. 120–127. doi: 10.1109/ICCCRI.2016.26.

Shah, R. *et al.* (2016) ‘Review Paper on Overloading Effect’, *International Journal of Advanced Scientific Research and Management*, 1(4).

Shimshoni, Y. *et al.* (2015) ‘Effects of parental vigilant care and feedback on novice driver risk’, *Journal of Adolescence*. Elsevier Ltd, 38, pp. 69–80. doi: 10.1016/j.adolescence.2014.11.002.

Simons-Morton, B. G. *et al.* (2013) ‘The Effect on Teenage Risky Driving of Feedback From a Safety Monitoring System: A Randomized Controlled Trial’, *Journal of Adolescent Health*, (53), pp. 21–26. doi: 10.1016/j.jadohealth.2012.11.008.

Specht, D. F. (1991) ‘1991_ITNN_A general regression neural network.pdf’, *IEEE Transaction on Neural Network*, pp. 568–576.

Stoichkov, R. (2013) ‘Android Smartphone Application for Driving Style Recognition’, *Lehrstuhl fur Medientechnik, Technische*.

Strobl, C., Malley, J. and Gerhard Tutz (2009) ‘Characteristics of Classification and Regression Trees, Bagging and Random Forests’, *Psychol Methods*, 14(4), pp. 323–348. doi: 10.1037/a0016973.An.

Swinscow, T. (1997) ‘Statistics at Square One: 11. Correlation and regression’, *British Medical Journal*.

Takeda, H. (1994) ‘Abduction for design’, *Formal Design Methods for CAD*.

Takeda, H., Veerkamp, P. and Yoshikawa, H. (1990) ‘Modeling Design Process’, *AI Magazine*, 11(4), pp. 37–37. doi: 10.1609/AIMAG.V11I4.855.

Takeda, K. *et al.* (2011) ‘Improving driving behavior by allowing drivers to browse their own recorded driving data’, in *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 44–49. doi: 10.1109/ITSC.2011.6082893.

Takeda, K. *et al.* (2012) ‘Self-coaching system based on recorded driving data: Learning from one’s experiences’, *IEEE Transactions on Intelligent Transportation Systems*, 13(4), pp. 1821–1831. doi: 10.1109/TITS.2012.2205917.

Tijerina, L. *et al.* (2000) *Preliminary Studies in Haptic Displays for Rear-end Collision Avoidance System and Adaptive Cruise Control System Applications*.

Toledo, T., Musicant, O. and Lotan, T. (2008) ‘In-vehicle data recorders for monitoring and feedback on drivers’ behavior’, *Transportation Research Part C: Emerging Technologies*, 16(3), pp. 320–331. doi: 10.1016/j.trc.2008.01.001.

Tolouei, R., Maher, M. and Titheridge, H. (2013) ‘Vehicle mass and injury risk in two-car crashes: A novel methodology’, *Accident Analysis and Prevention*. Elsevier Ltd, 50, pp. 155–166. doi: 10.1016/j.aap.2012.04.005.

Tong, S. *et al.* (2016) ‘Provision of telematics research’, p. 76p.

Tselentis, D. I., Yannis, G. and Vlahogianni, E. I. (2017) ‘Innovative motor insurance schemes: A review of current practices and emerging challenges’, *Accident Analysis and Prevention*. Elsevier Ltd, 98, pp. 139–148. doi: 10.1016/j.aap.2016.10.006.

Vaiana, R. *et al.* (2014) ‘Driving behavior and traffic safety: An acceleration-based safety evaluation procedure for smartphones’, *Modern Applied Science*, 8(1), pp. 88–96. doi: 10.5539/mas.v8n1p88.

Vaishnavi, V. and Kuechler, B. (2004) *Design Science Research in Information Systems*. Available at: http://desrist.org/desrist/content/design-science-research-in-information-systems.pdf (Accessed: 21 December 2017).

Vlahogianni, E. I. and Barmpounakis, E. N. (2017) 'Driving analytics using smartphones: Algorithms, comparisons and challenges', *Transportation Research Part C: Emerging Technologies*. Elsevier Ltd, 79, pp. 196–206. doi: 10.1016/j.trc.2017.03.014.

Wahlstrom, J., Skog, I. and Handel, P. (2014) 'Risk assessment of vehicle cornering events in GNSS data driven insurance telematics', in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, pp. 3132–3137. doi: 10.1109/ITSC.2014.6958194.

Wahlstrom, J., Skog, I. and Handel, P. (2017) 'Smartphone-Based Vehicle Telematics: A Ten-Year Anniversary', *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–24. doi: 10.1109/TITS.2017.2680468.

Wahlström, J., Skog, I. and Händel, P. (2014) 'Detection of Dangerous Cornering in GNSS Data Driven Insurance Telematics', *IEEE Transactions on Intelligent Transportation Systems*.

Wahlström, J., Skog, I. and Händel, P. (2015) 'Driving Behavior Analysis for Smartphone-based Insurance Telematics', in *Proceedings of the 2nd workshop on Workshop on Physical Analytics - WPA '15*. New York, New York, USA: ACM Press, pp. 19–24. doi: 10.1145/2753497.2753535.

Wahyudi, W., Ganis, E. and Taufik Mulyono, A. (2014) 'Effect of Overloading Freight Vehicles to Increased Carbon Dioxide Emissions (Case Study: East Java Province)', *International Refereed Journal of Engineering and Science (IRJES) ISSN (Online)*, 3(6), pp. 2319–183.

Wampler, D. (Lightbend) (2018) *Fast Data Architectures for Streaming Applications*. Second. Edited by J. Hassell, J. Billing, and R. Monaghan. O'Reilly.

Wieringa, P. R. (2016) *Design science research in information systems and software systems engineering Research methodology accross the disciplines*.

Wieringa, R. (2014) 'Empirical research methods for technology validation: Scaling up to practice', *Journal of Systems and Software*. doi: 10.1016/j.jss.2013.11.1097.

Wieringa, R. and Daneva, M. (2015) 'Six strategies for generalizing software engineering theories', in *Science of Computer Programming*. doi: 10.1016/j.scico.2014.11.013.

Wieringa, R. J. (2014) *Design science methodology: For information systems and software engineering*, *Design Science Methodology: For Information Systems and Software Engineering*. doi: 10.1007/978-3-662-43839-8.

World Health Organization (WHO) (2015) *International Traffic Safety Data Systems Improvement*.

Wu, M., Zhang, S. and Dong, Y. (2016) 'A Novel Model-Based Driving Behavior Recognition System Using Motion Sensors.', *Sensors (Basel, Switzerland)*. Multidisciplinary Digital Publishing Institute (MDPI), 16(10). doi: 10.3390/s16101746.

Xu, S. and Zhao, Q. (2011) 'Study on Vehicle-mounted Overloading Control System for Passenger Vehicles', *Procedia Engineering*, 15, pp. 1214–1218. doi: 10.1016/j.proeng.2011.08.224.

Yan, X., Wang, J. and Wu, J. (2016) 'Effect of In-Vehicle Audio Warning System on Driver's Speed Control Performance in Transition Zones from Rural Areas to Urban Areas.', *International journal of environmental research and public health*. Multidisciplinary Digital Publishing Institute (MDPI), 13(7). doi: 10.3390/ijerph13070634.

YAO, X. (1999) ‘Evolving artificial neural networks with FINCH’, *PROCEEDINGS OF THE IEEE*, 87(9), p. 1719. doi: 10.1145/2464576.2480780.

Zabihi, H. *et al.* (2017) ‘WatchOut: Facilitating Safe Driving Behaviors with Social Support’, in *CHI’17 Extended Abstracts,* pp. 17–5. doi: 10.1145/3027063.3053188.

Zhao, H. *et al.* (2013) ‘Join driving: A smart phone-based driving behavior evaluation system’, *GLOBECOM - IEEE Global Telecommunications Conference*. IEEE, pp. 48–53. doi: 10.1109/GLOCOM.2013.6831046.

Zheng, Y., Member, S. and Hansen, J. H. L. (2016) ‘Unsupervised Driving Performance Assessment using Free- Positioned Smartphones in Vehicles’, in *19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1598–1603.