

Research Article

Design and Development of Weigh-In-Motion Using Vehicular Telematics

Sivaramalingam Kirushanth  and Boniface Kabaso

Department of Information Technology, Faculty of Informatics and Design, Cape Peninsula University of Technology, Cape Town 8000, South Africa

Correspondence should be addressed to Sivaramalingam Kirushanth; sivaramalingamk@vau.jfn.ac.lk

Received 24 July 2019; Revised 7 December 2019; Accepted 26 December 2019; Published 4 March 2020

Academic Editor: Xinyu Liu

Copyright © 2020 Sivaramalingam Kirushanth and Boniface Kabaso. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Identifying overloaded vehicles on a highway is essential for the safety of vehicles on the road as well as for the performance monitoring of highway infrastructure and planning. Traffic enforcement uses various weigh-in-motion (WIM) methods. Since Vehicular Telematics (VT) is favoured in the transport industry, using it for building a new WIM system to infer the payload of a vehicle at any road segment would be beneficial for the transport industry. This paper presents the effort taken to use VT data from onboard diagnostics modules and smartphones to infer the payload of a vehicle. The experiment done to find the correlation between VT data and the payload of a vehicle is discussed. Feature engineering was done; nine different settings were tested to find the best regression model. A multiple nonlinear regression model produced significant a p value of $6.322e-08$ and an R -squared value of 0.8736. Results support the notion of using the VT data for nonintrusive measurement of the weight of a vehicle in motion.

1. Introduction

Road safety is one of the most significant issues in the world [1]. Driving an overloaded vehicle causes various kinds of hazards such as mechanical failures and structural deformation of vehicles and roads, which lead to accidents, and it is an illegal and punishable offence in most countries. According to the South African National Road Traffic Regulations, driving an overloaded vehicle leads to prosecution for an offence under regulations in the National Road Traffic Act, 1996 [2]. According to the U.S. Department of Transportation, vehicle condition and road/environment conditions are the two factors which are collectively responsible for 5.2% of road accidents [3]. Vehicles carrying more than the manufacturer's specified and permitted payload are considered overloaded. In other words, a vehicle is overloaded if the total weight of a vehicle when fully loaded is more than the maximum allowed Gross Vehicle Weight (GVW), where the GVW is the sum of kerb weight and payload [4].

Weighing the weight of a moving vehicle on the road is known as weigh-in-motion (WIM). Fred Moses and George

Globe introduced Bridge WIM (B-WIM) in the USA in the early 1970s. The successful B-WIM application took place in Australia in the mid-1980s [5]. WIM has been used in the transport industry for more than a decade and for many reasons. Earlier, it was only used to plan and build the roads and bridges. In recent years, the legislation has been changed, and the WIM data is also used by traffic enforcement departments for the enforcement of overloading. Identifying an overloaded vehicle driving on any road is still a tough task for enforcement officials. In many countries, the high-speed WIM (HS-WIM) is used to detect the overloaded vehicle on the road; the selected vehicles are then screened on a static WIM to obtain more accurate weight. The present HS-WIMs have the accuracy of 5%-15% due to various internal and external disturbing factors [6].

The vehicle industry has used Vehicular Telematics (VT) for more than a decade for various reasons. Pay As You Drive (PAYD) or User-Based Driving Insurance (UBI) is the most popular insurance schemes used by vehicle insurance companies all over the world. On-board devices are installed on the user vehicles to collect driving information as they drive.

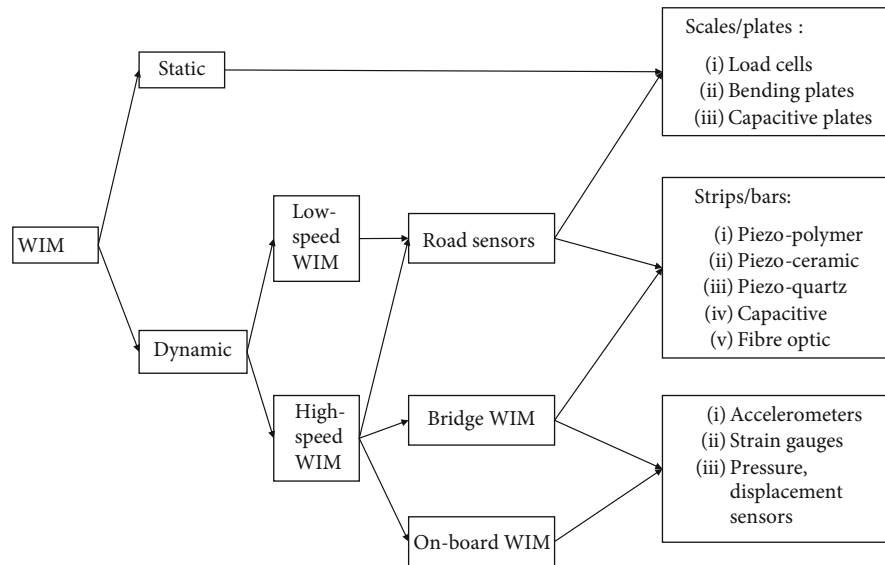


FIGURE 1: Classification of WIM [9].

Installing such devices is becoming mandatory in some countries [7]. These VT data are used to analyze the driving behaviour and road anomalies. The availability of such data offers more paths for further research. Even various kinds of WIM solutions are being used in practice, each of which has its advantages and disadvantages. Machine learning (ML) algorithms are widely used in Intelligent Transportation Systems (ITS) and its applications. A new WIM solution using VT and ML was proposed in [8]. This paper illustrates the prototype design of the proposed WIM system and discusses the results obtained from it. The next section discusses the available WIM systems, ability of VT data, and ML. The following section describes the prototype design considerations and solutions used. Various features were tested to find a better regression ML model. Finally, the prototype system is compared with other WIM systems.

2. Background

In general, WIM systems are used to measure the GVW and other parameters of vehicles [6]. Two main classifications of WIM solution methods are static WIM and dynamic WIM. There are two subcategories of dynamic WIM methods which are (1) low-speed WIM (LS-WIM) and (2) high-speed WIM (HS-WIM). Figure 1 shows the classification of WIM by [9]. In LS-WIM, the vehicle is weighed while it moves across the scale at low speed, typically less than 15 kmph, but HS-WIMs are capable of weighing the vehicle weight at full highway speeds [5].

2.1. Static WIM. In a static WIM method, the vehicle is weighed while it is stationary on the scale. Static WIM methods are mostly accurate but cumbersome. Fixed systems, semiportable systems, and portable systems are the three types of static WIMs in general [5]. Fixed systems are permanently mounted to the pavement, usually in a reinforced concrete frame or platform. Semiportable 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 [5].

2.2. Low-Speed WIM. According to [6], 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 30-40 m in length. The vehicle may be guided by curbs to minimize variation in the transverse position of the wheels. The data processing system analyzes 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 [6].

2.3. High-Speed WIM. HS-WIM systems are built to measure the weight of vehicles driving in 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 installed either 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 [5].

The multisensor WIM (MS-WIM) was introduced in [6], because of the low accuracy or correctness of HS-WIMs. The

authors of [6] 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 major concern. In the report [10], researchers identified three major factors that affect the accuracy of the WIM system, which are 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 [10]. They also reported that vehicle characteristics, such as speed, tyre type, inflation pressure, suspension system, and axle configurations, affect the dynamic tyre force, thus affecting WIM sensor accuracy. According to [6], 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 [6]. 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. In [11], a WIM by observing the length of the shock absorber was developed in two-wheeler vehicles. Reference [12] discusses an experiment to explore the various possibilities for a passive WIM system. Researchers in [12] investigated multiple vehicle indicators including brake temperature, tyre temperature, engine temperature, acceleration and deceleration rates, engine acoustics, suspension response, tyre deformation, and vibrational response. Their sensing system included infrared video cameras, triaxial 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 [13] 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 driving force is calculated by the torque value, and the acceleration is calculated by the speed value. Present vehicles use Electronic/Engine Control Unit (ECU) 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 [14].

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. 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.

2.4. Vehicular Telematics. VT is a system that comprises the various sensory devices, communication methods and the applications using the data. The VT data is also referred to as flying car data. It has been used in the transport industry for more than a decade. PAYD or UBI rely heavily on using this data. Onboard diagnostic modules or a black box is used

to collect this data from Engine Control Units via a Controller Area Network (CAN) Bus. In addition to that, an Inertial Measurement Unit (IMU) is also used. Driving behaviour detection and road anomaly detection is the primary application of this data. From the literature studies, VT data collection devices are available at little or no cost in cases where it is mandatory for insurance purposes and legislations in some countries [15–17]. Table 1 summarizes some of the available VT data collection devices, also known as black boxes as in 2017 adapted from [18].

Studies reveal that there are several methods and data used in driving behaviour and road anomaly detection [19]. IMUs such as Accelerometer and Gyroscope are some of the most widely used sensory devices in research. Accelerometers measure the 3-dimensional acceleration force applied to the observing device. Gyroscope measures the Pitch, Yaw and Roll (3-Dimensional Rotation) of an observing device. Global Navigation Satellite Systems (GNSS) are used to find the position of the device on the globe (earth) using GPS and/or GLONASS positioning satellites. The latitude, longitude and altitude at a point in time are observed and captured from these GNSS devices. The precision of the reading depends on the devices and the connected number of satellites.

The introduction of Onboard Diagnostics (OBD) port on modern vehicles manufactured after 1996 has enabled us to read the CAN Bus data. The CAN bus data brings much information from the ECU. Information such as revolution per minute (RPM), throttle position, intake air temperature, coolant temperature, oil pressure and many more other manufacturer specific data can be obtained from the vehicle by connecting the OBD Dongle. The data can be transmitted via a wired or wireless (Bluetooth/Wi-Fi) medium.

2.5. 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.

The ECU receives the values from the array of sensors and interprets the values with a multidimensional performance map and controls the actuators accordingly [20]. 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 is one of the safety functions with regard to safety. Numerous sensors are being used in autonomous vehicles (see Figure 2). Table 2 contains some of the control systems, sensors, and actuators in automobile vehicles.

2.6. Controller Area Network (CAN) Bus Data. The sensor data transmission happens via the CAN bus to the ECU. With a large number of components that exchange data through a technology invented in 1986 by Robert Bosch [21], a serial broadcast bus allows near real-time management of most sensors and electronic devices embedded in the car [20]. The CAN bus transmits ECU data outside for troubleshooting and performance logging using several standards, e.g., J1939.

TABLE 1: Some of the telematic devices and their cost [18].

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

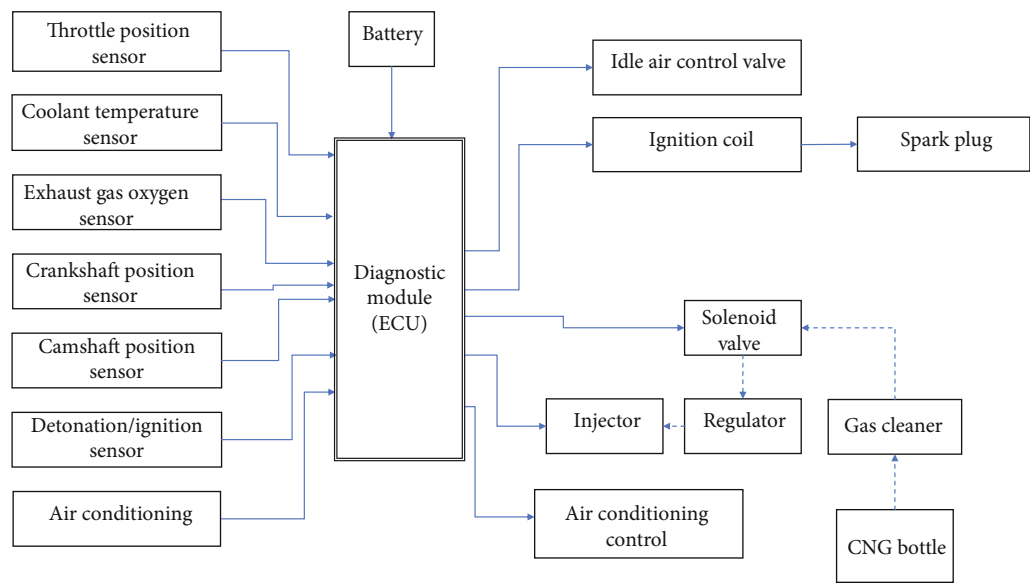


FIGURE 2: Schematic diagram of ECU with some sensors and actuators.

TABLE 2: Example of some automotive control systems [20].

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 electrochemical	Fuel injector
Knock control	Knock	Knock sensor output	Ignition timing	Piezoelectric accelerometer	Ignition coil switch transistor
Antilock braking system	Wheel slip limit	Wheel speed	Brake time pressure	Magnetic reluctance	ABS solenoid valve

2.7. Onboard Diagnostics Module. Almost all of the newly produced automobiles are required, by law, to provide an interface for the connection of diagnostic test equipment [22]. Cars manufactured after 1996 have an 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. ELM 327 acts a bridge between OBD ports and a standard RS232 serial interface. The ELM microchip has its interface to send and receive data from external applications, and the ECUs are designed by various manufacturers.

Figure 3 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 the 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, ELM 327 tries to establish a connection with the vehicle’s ECU. It tries to establish the connection with several communication protocols and baud rates. According to the ELM electronics report [22], there are 12 different protocols supported by ELM,

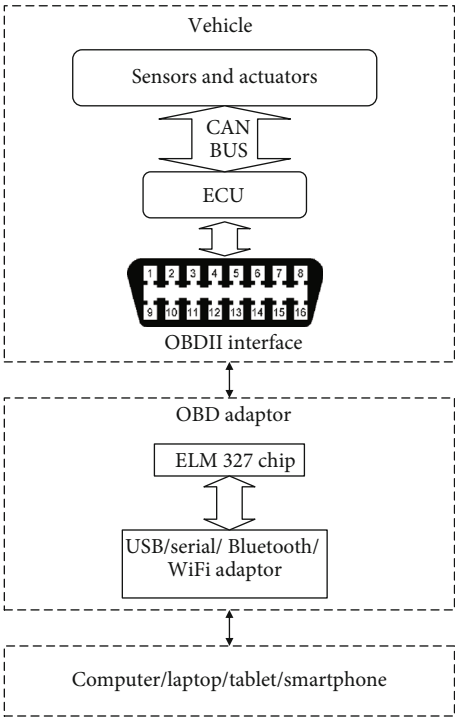


FIGURE 3: Block diagram showing ECU data collection modules.

TABLE 3: 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)

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 3 describes the ten diagnostic services described in the latest OBD-II standard Society of Automotive Engineers (SAE) J1979. Before 2002, J1979 referred to these services as “modes”.

The parameters are accessed using their Parameter Identifiers (PIDs); for example, Engine Revolutions Per Minute (RPM) has PID number 12 under service number one. 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 heavy-duty vehicles [16].

In [23, 24], the speed and acceleration were calculated only using GNSS data. They reported that the backward difference method to find speed and acceleration using GNSS data had shown more accuracy. A combination of GNSS and IMU data was used in [3] to detect dangerous cornering events. A study [25] solely used OBD-II data to detect reckless driving behaviour and vehicle anomalies. A similar study [26] used speed, acceleration, and engine RPM as parameters for driving behaviour detection. Reference [23] reports that under controlled experiments, the total accuracy of 99.5% was achieved when using smartphone sensors (IMU and GNSS) and 99.3% was achieved when using the OBD-II device. Gyroscope, accelerometer, GNSS, and microphone sensor data were used in [24] for a turning and cornering detection system. The system was deployed on smartphones. The microphone was used to detect the signal relay sound. The study [27] used several sensors to collect rich multimodal data. 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 are some

of the data gathered in this research to study the driving behaviour. A recent study [28] investigating several smartphone sensors and ML algorithms for driving behaviour and road anomaly detection found that the accelerometer and gyroscope are the best sensors to detect driving behaviour. In studies [24, 29, 30], sensor fusion was found more promising in driver behaviour and road anomaly detection. Article [31] 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 II data in this project.

2.8. Theoretical Background. According to Newtonian 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 a 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.

In vehicles powered with 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 air flow. According to [12], the weight of a vehicle can be measured using several internal and external features. In [13], the estimating means of vehicle weight is based on the motion and is given by the following equation:

$$ma = F - (m \times g \times \sin(\Theta)) - R, \quad (1)$$

where a is the acceleration in ms^{-2} , m is the vehicle mass in kg, F is the driving force in N, Θ is the slope of the driving surface in degrees, g is the gravitational acceleration in ms^{-2} , and R is the running resistance in N.

From Equation (1), the mass is

$$m = \frac{F - R}{(a + g \times \sin(\Theta))}. \quad (2)$$

Lin and Li [32] 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 tyre, (6) air pressure of one or more tyres, (7) air resistance, (8) size and shape of the vehicle, (9) alignment of wheels, and (10) transmission system. The 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 (3), which is typically an indication of the current airflow

divided by the peak airflow at the wide open throttle as a function of RPM, where airflow is corrected for altitude and ambient temperature [33].

$$EL = \frac{CAFR}{PAFR \times (BARO/29.92) \times \sqrt{298/(AAT + 273)}}, \quad (3)$$

where CAFR is the current air flow rate, PAFR is the peak air flow rate at fully open throttle at standard temperature (25°C) and pressure (29.92 in Hg BARO), and AAT is the ambient air temperature (in °C).

In summary, force (F) applied on an object with mass (M) produces an acceleration (A) (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 (3). From these two factors, we could say that a vehicles' EL is proportional to the weight of it at a certain acceleration. But EL is influenced by several internal (Equation (3)) and external (Equation (2), [20, 21, 34]) factors. We assume that the relation is multiple linear regression. It can be viewed as

$$W = b + \sum_{i=0}^n a_i \times x_i, \quad (4)$$

where the weight of a vehicle W is the sum of a bias value b and the accumulated sum of the products of coefficient a_i and feature x_i of all the n number of features.

2.9. Machine Learning. ML is a form of Artificial Intelligence (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:

$$T = (y_n, x_n), \quad n = \{1, \dots, N\}, \quad (5)$$

where y_n are the response (dependent) vectors and x_n are the 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 (y_n, x_n) are independent and identically distributed variable vectors from the distribution (Y, X) and given 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 is the following:

$$PE(f, T) = E_{Y,X} L(Y, f(X, T)). \quad (6)$$

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 predicted category, or the label is not equal to the true (given) label, zero otherwise.

2.10. 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 [35].

2.11. Machine Learning 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. In supervised learning, 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 creates models from a set of inputs with no labelled examples. 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.

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 and 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).

2.12. 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 .

$$y = \beta_0 + \sum_{n=1}^N (\beta_n x_n) + \varepsilon, \quad (7)$$

where β_0 is the intercept term and 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 the mean

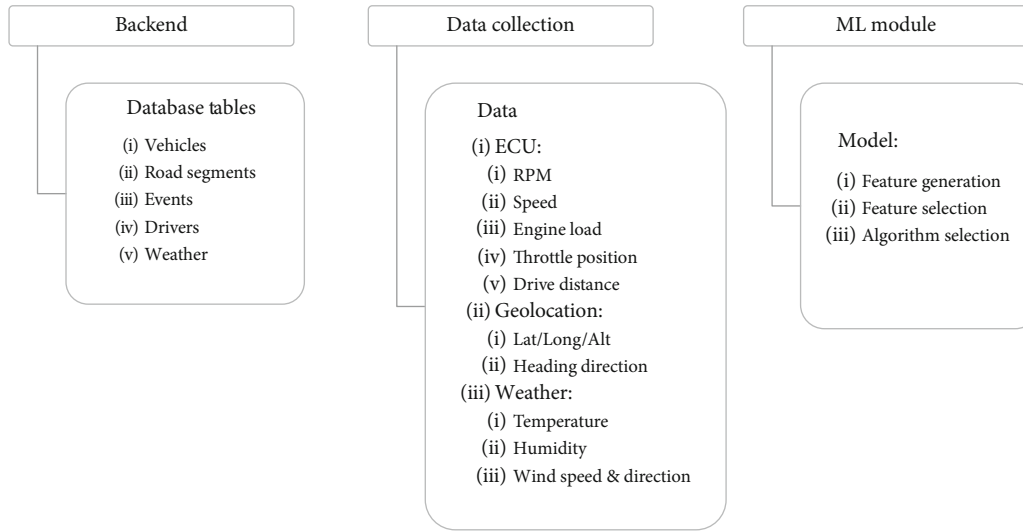


FIGURE 4: Overview of the proposed WIM system components (data/functions).

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

$$\text{RSS}(\beta) = \sum_{n=1}^N (y_n - \hat{y}_n)^2. \quad (8)$$

The coefficients (β) which make the smallest RSS/SSE value are obtained from the maximum likelihood estimate of β . This way of fitting the model by minimizing the RSS is called Ordinary Least Squares (OLS) [29].

Let $Y = (y_1, \dots, y_N)^T$ 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

$$\hat{\beta} = (X^T X)^{-1} X^T Y. \quad (9)$$

OLS fit methods work well for single independent variable and single dependent variable regressions. If the response variable is in a nonlinear relation with more than one predictor variable, the relation is called multiple nonlinear regression or, in some cases, multiple polynomial regression. It is simply achieved by introducing new variables by applying nonlinear functions such as log, sin, and square root, to the existing predictor variables. The 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 [30].

2.13. Problem Definition. VT data is becoming more abundant and more accurate due to the technological improvements in sensors and connectivity. ML approaches are widely used in identifying driving behaviour and road anom-

aly detection [28] and also in other WIM solutions [36]. Building a new WIM system to infer the payload of a vehicle on any road segment using VT data and ML would help the transport industry. Exploring the feasibility of inferring the payload using VT and ML needs to be done through designing and testing the prototype.

3. Materials and Methods

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. The conceptual design framework of the WIM application was proposed considering three modules: the back-end module, the data collection module, and the ML module. The first two modules were independent, and the later module was dependent on the other two modules.

Figure 4 shows the order of the design and development performed, considering their dependencies. 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 and send the received data to the backend as stream or batch input. The ML module was designed to infer the weight of a vehicle caring (payload). ML models were chosen by training and testing with the data present in the backend. The WIM application was designed to use Application Programming Interfaces (APIs) to receive from a wide range of sources using a JavaScript Object Notation (JSON) format.

Figure 5 shows the architecture of the prototype to handle fast data handling from multiple VT data devices, depicted as the Internet of Vehicle (IoV) devices. The

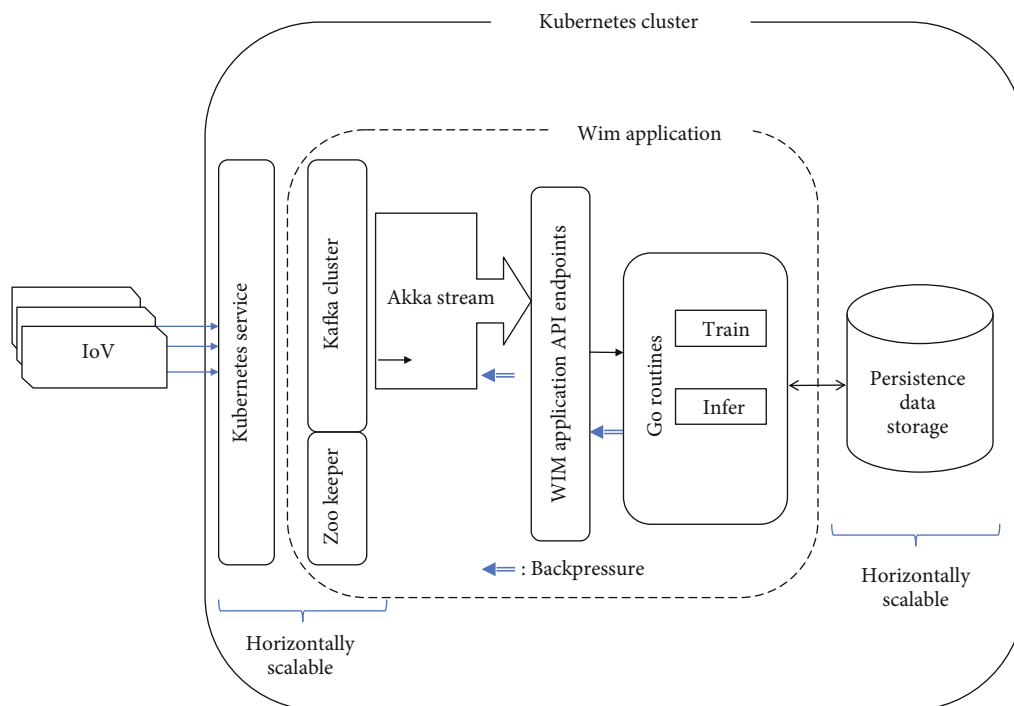


FIGURE 5: System architecture with stream processing [34].

prototype system application is deployed on a Kubernetes cluster of five physical nodes [34]. The system receives tons of VT data from each IoV device. 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 WIM application. The 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 processed by goroutines. Akka stream is used to stream each VT data (Kafka topics) for further processing. Kafka's "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, and results). This persistence storage is easily 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 to the Akka streams to the Kafka Cluster. This will trigger the streaming to be flexible with the backpressure by reducing the streaming data rate. But this could lead to the Kafka Cluster overflowing by the fast-incoming VT data accumulation. In such cases, the Kafka Cluster with the help of Zookeeper could scale up horizontally.

3.1. 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 orchestra-

tion engine which runs and manages Linux containers. Kubernetes is an open source platform for automating deployment, scaling, and operations of application containers across clusters of hosts, providing container-centric infrastructure [37]. The application was containerised and deployed in Kubernetes cluster with five physical nodes. Persistence storage was provided by the three Gluster [38] nodes, including one Kubernetes node. Kubernetes container orchestration engine runs several stateful sets, 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 sets. In this research, GlusterFS is used to maintain the persistence volumes for the Kubernetes cluster. GlusterFS is a distributed, software-defined file system, where storage devices, called "bricks," are selected on one or more nodes to form logical storage volumes across a Gluster cluster. It is easy to increase storage by simply adding more nodes, which provides features like cross-node and cross-site replication, usage balancing, and iSCSI storage access [38]. 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.

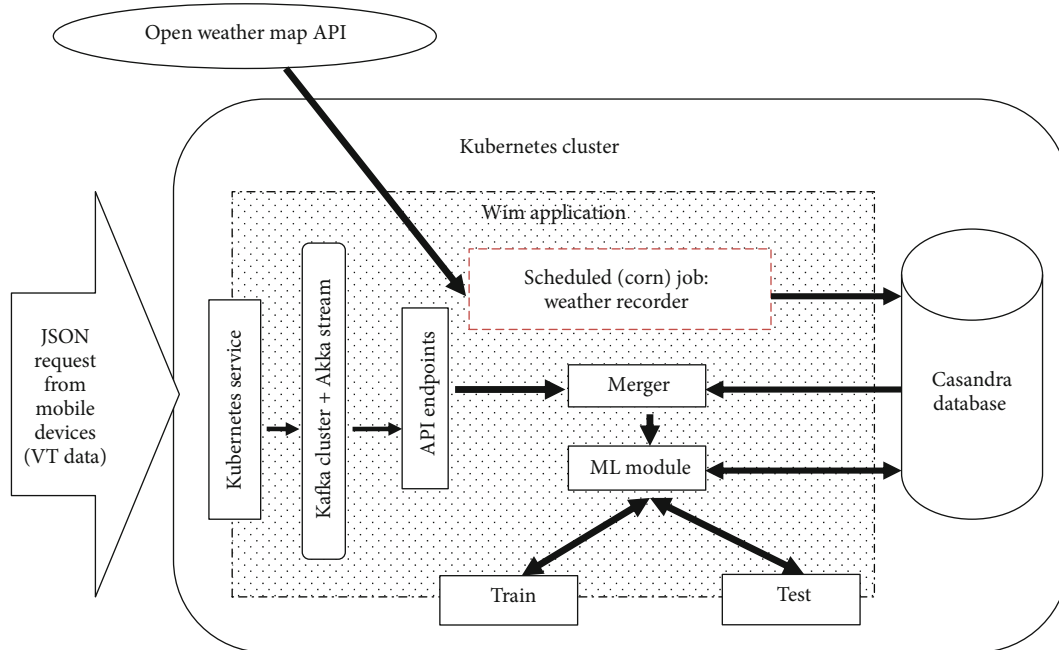


FIGURE 6: High-level graphical structure of the developed WIM application [34].

3.2. Languages. Golang (Go) was chosen as the programming language to develop the ML module of the WIM Application. Scala was used to build the Akka streaming API 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 machine learning.

3.3. 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 weather information from a scheduled job (cron), to retrieve stored VT data and weather data to train and test ML models, and to store the inferred output.

Figure 6 portrays the application architecture of the developed conceptual framework focused on the VT data ingestion. The WIM application has two main jobs, a scheduled weather data recorder and an ML Application. A scheduled job to read the current weather information for the selected places from the OpenWeatherMap was deployed. 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 in 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 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 cron job automatically collected current weather data of the prior set locations from the OpenWeatherMap and stored it into the Cassandra database.

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 the time value. The result of the inference is stored in the Cassandra database. The stored inferred results could then be served to any frontends upon request.

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

- (i) an OBD-II Bluetooth/Wi-Fi module
- (ii) an android mobile device
- (iii) a WIM inference engine application running on a Kubernetes cluster

The OBD-II module was used to collect the CAN bus data. An android mobile device was used to fetch CAN bus data from the OBD-II module via Bluetooth or Wi-Fi. The android mobile device collected ECU data for each second and stored it along with the data from the built-in GNSS position data. The collected/stored data was then sent to the WIM API server using the REST clients through the service API endpoints. Figure 7 shows the schematic diagram of

TABLE 5: Details of the variables collected during the data collection.

Variable	Description	Source	Range	Units
ACC	Acceleration	$= (\Delta VS * 0.27) / \Delta t$	(-10)-10	ms-2
ALT	Altitude (meters above sea level)	GNSS	10000+	m
LAT	Latitude	GNSS	41.25-(-120.9762)	—
LON	Longitude	GNSS	-31.96-(-115.84)	—
ELE	Elevation	$= \arctan (\Delta LAT / \Delta DD)$	($\pm$)45	Degrees
VS	Vehicle speed	OBD/GNSS	0-220	kmph
DD	Drive distance	OBD	0-10,00,000	km
EL	Calculated engine load	OBD	0-100	%
RPM	Revolutions per minute	OBD	0-9000	r/min
TP	Throttle position	OBD	0-100	%
HUM	Humidity	OpenWeatherMap	0-100	%
TEM	Temperature	OpenWeatherMap	0-350+	K
PRE	Atmospheric pressure	OpenWeatherMap	1000+	hPa
WS	Wind speed	OpenWeatherMap	0-400+	kmph
WD	Wind direction	OpenWeatherMap	0-360	Degrees
Weight	Payload	User input	50-650	kg

Technology (CPUT). Volunteers weighing different weights participated as passengers during data collection. The car was driven only in the first and second gears. The landscape of the University parking 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 5 kmph. The uncontrolled data was collected from the daily commuting of the 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 driver with the mass of any bags carried). Since the density of the fuel is 0.7 kg/l and the fuel tank capacity is 43 litres, it makes a significant 30 kg difference in total weight. The weight of the fuel was also considered in four-quarter blocks by observing the fuel gauge reading.

3.6. 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 = 1 Hz).

Table 5 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), and altitude (ALT) were collected from the smartphone's GNSS unit. The combined data with the time-stamp and the geolocation was then used to extract the weather information from the stored weather database since EL depends on airflow, standard temperature, and pressure. To reduce multicollinearity, airflow, standard temperature, and pressure readings were not recorded nor included in the feature set.

3.7. Correctness of Data

3.7.1. 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 essential data for our inference system. 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 one hour [20]. The direction of the wind and the speed may vary due to the landscape and the objects. The resolution of our weather data obtained was two hours. Most of the data recorded have remained unchanged, or data with minimal variance. 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.

3.7.2. Vehicle Speed from ODB vs. GNSS. The VS collected from the OBD vs. GNSS is shown in Figure 8. Pearson's product-moment correlation coefficient (PPMCC) [40] was used to check the correlation between the two different readings. PPMCC between two vectors: $X = \{x_1, \dots, x_N\}$ and $Y = \{y_1, \dots, y_N\}$, is

$$\text{PPMCC} = \frac{S_{xy}}{\sqrt{S_{xx}S_{yy}}}, \quad (10)$$

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

The PPMCC of the VS readings from OBD and GNSS is 0.842. The zero readings for nonzero values of speed OBD

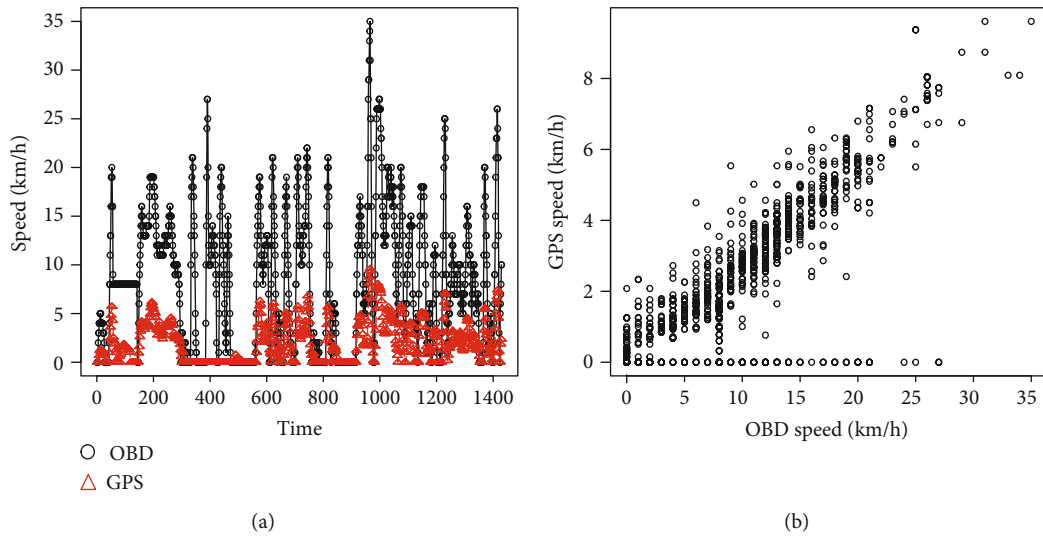


FIGURE 8: Difference in speed reading from OBD and GPS. (a) Speed readings. (b) Correlation graph.

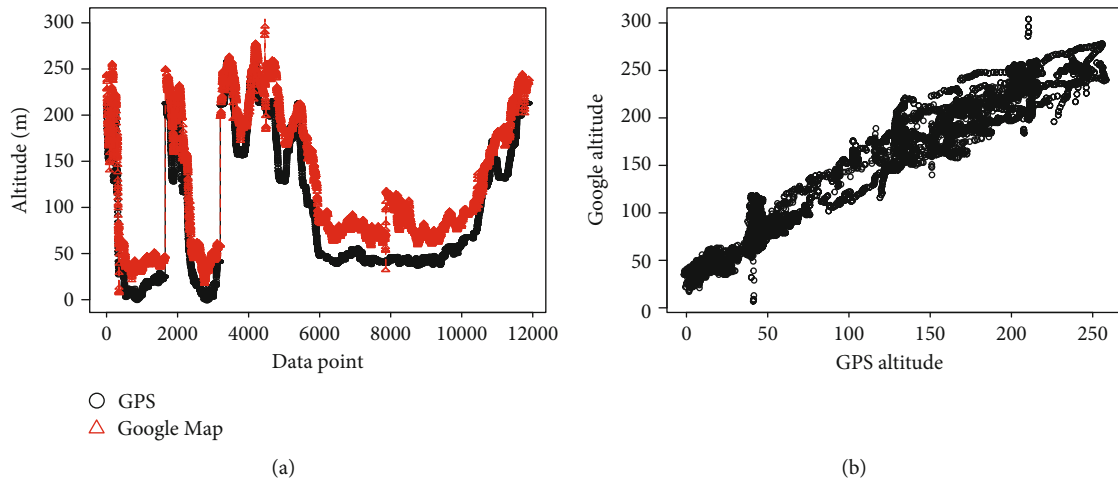


FIGURE 9: Difference in altitude from GPS and Google Map from uncontrolled driving data. (a) Altitude readings. (b) Correlation graph.

readings are due to the time taken to fix GNSS satellites for positioning. Due to this reason, the speed in this research was chosen from the OBD reading.

3.7.3. Altitude Form GNSS vs. Google Map API. Altitudes from Google Map API for a list of latitude and longitude positions bring a different value from GNSS sensor altitude, as shown in Figure 9. The PPMCC between these two altitude measurements is 0.976.

3.8. Road Gradient (Elevation Angle). The phone's rotation sensor was tested to be used to find the ELE of the road. In order to obtain the ELE, 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/deacceleration (ACC) of the vehicle due to throttling and braking. Equations (11) and (12) were used to calculate ACC and ELE, respectively.

$$ACC = \frac{\Delta VS \times 1000}{\Delta t \times 60 \times 60} \text{ms}^{-2}, \quad (11)$$

where ΔVS is the change of vehicle speed in kmph and Δt is the change of time in seconds.

The following is the road gradient/elevation angle in degrees:

$$ELE = \tan^{-1} \left(\frac{\Delta ALT}{\Delta DD \times 1000} \right), \quad (12)$$

where ΔALT is the change of altitude m and ΔDD is the drive distance in km.

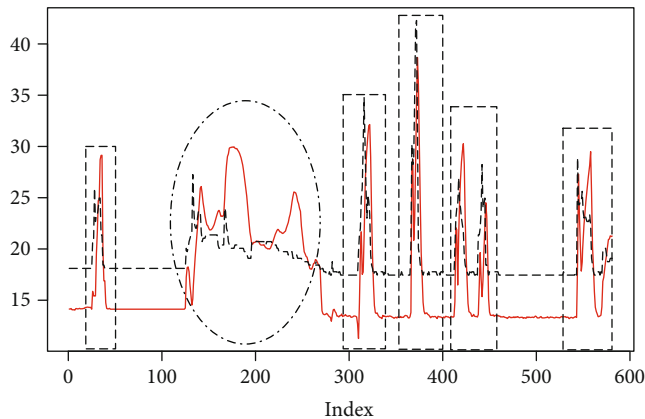


FIGURE 10: Throttle response time (throttle position in % and RPM).

3.9. 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.

In Figure 10, 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 TP (i.e., ΔTP), 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 (ΔTP) in normal conditions, that is, when either clutch is engaged (pedal released) and accelerating or when the clutch is disengaged. The delay in engine response in those regions is clearly visible. It was found that there is a 0.6-second 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. The frequency of parameters ranged from 1 Hz to 100 Hz but was collected at the rate of 1 Hz. The reduced rate of data collection might have missed some crucial facts from those data.

3.10. Data Preprocessing. 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 11 shows the values of RPM, VS, EL, and ACC of a journey for Point A to Point B within the 20s.

The first spike on the EL shows the gear change from the first gear to the second gear. During this period, the clutch will be released to separate the engine and transmission, TP will be decreased. Thus, the RPM will also be reduced. This speed difference (ΔVS) is very lean; therefore, the ACC reaches zero, then shoots up when the gear is changed. The graph segment between times greater than 15 depicts the braking (deacceleration) event to bring the vehicle to be stationary. ACC may occur due to two different reasons: (1) vehicle on an inclined or flat surface (i.e., $ELE \geq 0$) when TP is high, RPM is high, and EL is high and (2) vehicle on the declined surface (i.e., $ELE < 0$) when the TP, RPM, and

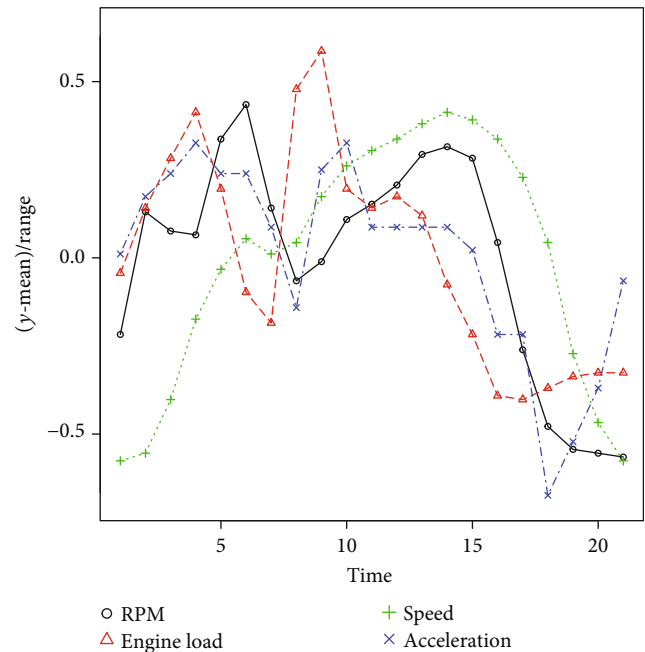


FIGURE 11: RPM, elevation, speed, and acceleration for a drive from Point A to Point B.

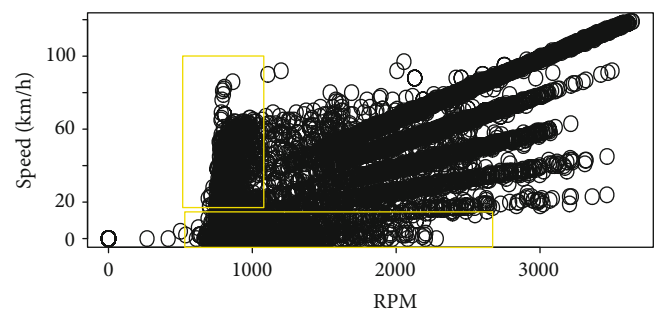


FIGURE 12: RPM vs. vehicle speed.

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., $ELE > 0$), with low TP, low RPM, and low EL and (2) on a flat or declined surface (i.e., $ELE \leq 0$) with high RPM, low TP, and low EL (usually on low gears) as explained by the oval shape in Figure 10. 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 $ACC \geq 0$ and $ELE \geq 0$ and $RPM > \text{minimum RPM}$ and $TP > \text{minimum TP}$.

Figure 12 shows the RPM vs. vehicle speed graph for the uncontrolled data collected on different payloads (95, 110, 112, 180, 240, and 320 kg). This graph clearly shows the correlation between VS 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 interested 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 VS gain during the first gear was captured for different payload settings. The left corner of 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.

3.10.1. Data Transmission. The collected VT data must be sent to the WIM system either in batch or in stream fashion. Streams with short bursts would be more preferable than a continuous stream of VT data. Assume a vehicle data collection device (sender) collects the VT data at a rate of 1 Hz and 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 from the VT devices should be minimized 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)
  THEN
    routeID ← VID + StartTime
    IF (speed < 20)
      THEN
        IF (connected)
          THEN
            Stream VT data
          ELSE
            Queue VT data
        ELSE
          Log Locationdata, routeID
      ELSE
        Connect and Send VT data queue, Log

```

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 0–20 kmph speed. 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 from 0 to 20 kmph) for each routeID. If the vehicle speed is greater than 20 kmph, 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.

Figure 13 describes the sample extracted data using the data extraction process. This extracted data was used to choose the ML model.

3.11. 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 not to violate laws or damage the testing vehicles. On the other hand, an attempt has been made to test the weight inference system using regression models.

3.12. Feature Creation. Feature engineering is the most difficult and time-consuming part of ML projects [41]. 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 preprocessing, the preprocessed 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 6 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.

The correlation between the independent variables is also known as multicollinearity. In here, the vehicle speed 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 vehicle speed is because RPM is less correlated to weight (0.05) than vehicle speed (0.12). Some new features were added by multiplying existing features and finding the powers of selected features. ACC, VS, RPM, EL, ELE, and TP are used to create new features using nonlinear functions such as $\text{Log}(x)$, $\text{Sqrt}(x)$, and $\text{Power}(x, -1)$, and $\text{Power}(x, 2)$, where $\text{Power}(a, b) = a^b$. Feature crossing is also done to obtain new features by multiplying and dividing existing features. n feature crossing is the combination of the multiplication of n features. Since we got the negative powers of features, the feature crossing results in inverse multiplication.

3.13. 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 are also affected by the number of features. It could also lead to model overfitting in some cases. Handling higher dimensional data is an issue with a higher feature space.

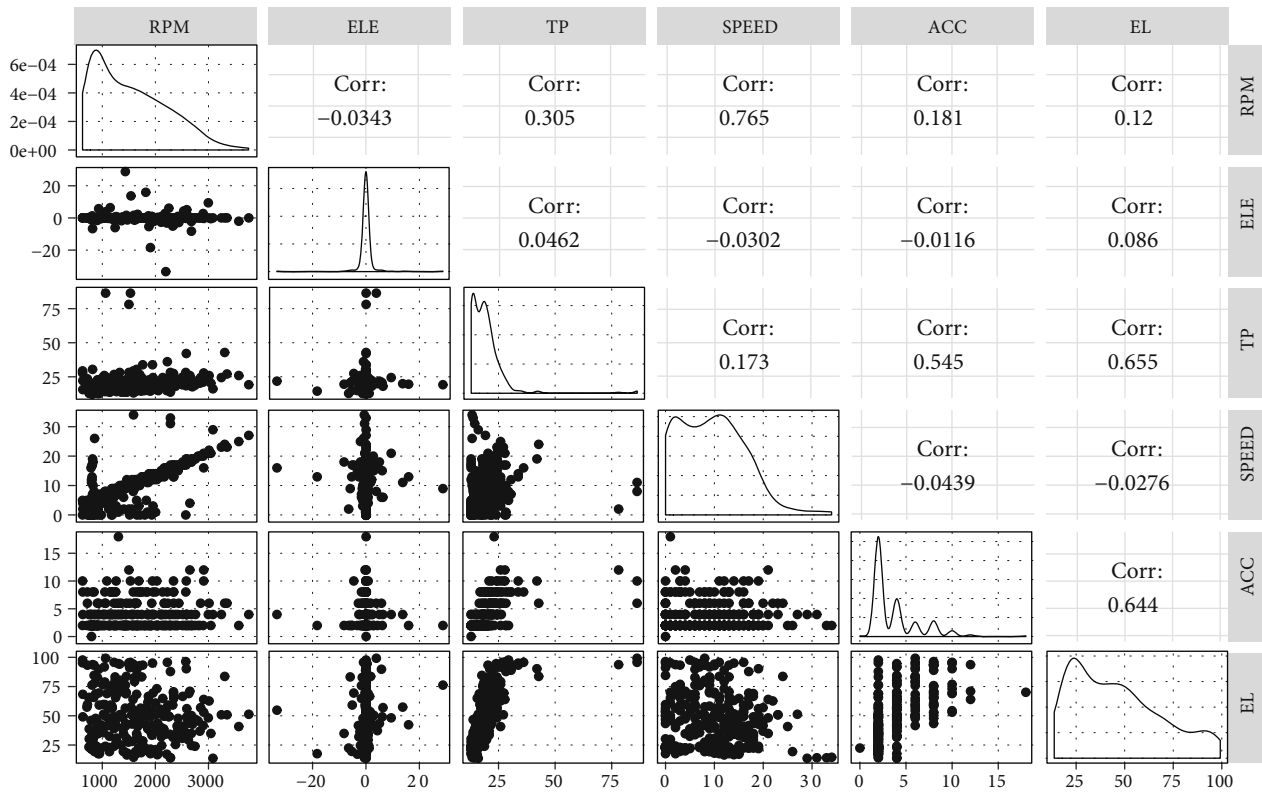


FIGURE 13: Scatter plot and correlation matrix of the primary features from the dataset.

TABLE 6: 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

There are several methods available for feature selection. Stepwise regression, penalised regression (ridge, lasso, and elastic), and principal component based regression [42] are some of the feature selection methods available. According to [42], 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) [43], showing that using the four-variable model results in the best RMSE value of the inferred weight. A stepwise feature selection based on Akaike Information Criterion (AIC) [43] was also performed. The results obtained using the stepwise regression are discussed under Results and Discussion. Nine different settings were made, and the performance was measured based on their standard residual 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 nonlinear functions such as $\text{Sqrt}(x_i)$, $\text{Power}(x_i, -1)$, $\text{Power}(x_i, 2)$, and $\text{Log}(x_i)$ to setting 1
- (6) Introducing new features by adding nonlinear functions such as $\text{Sqrt}(x_i)$, $\text{Power}(x_i, -1)$, $\text{Power}(x_i, 2)$, and $\text{Log}(x_i)$ to setting 3
- (7) Setting 6 with two feature crossing
- (8) Selecting the best features picked from 7 based on a significance value
- (9) Feature selection on setting 6 using stepwise AIC with 3 feature crossing

4. Results and Discussion

Table 7 summarizes the model performance results obtained from the previously mentioned nine settings.

TABLE 7: Results of the 9 settings.

Setting	Std residual 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

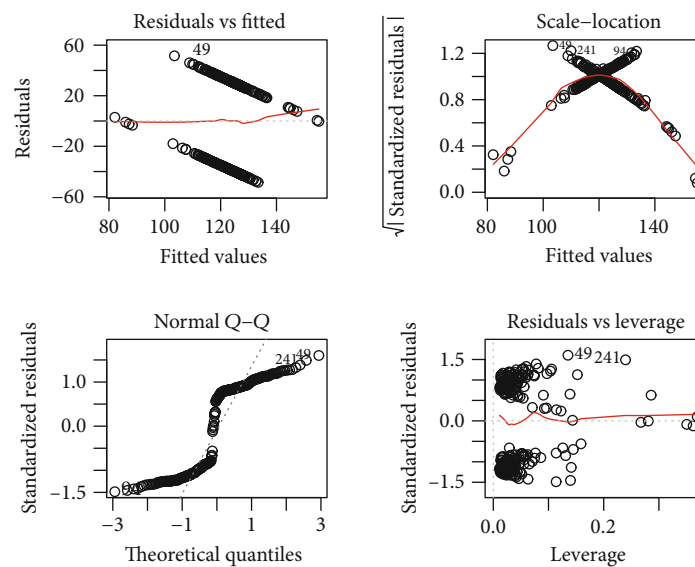


FIGURE 14: Setting 2 residual plots.

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. Setting 2 shows a better result than setting 1 with a smaller p value and an adjusted R -squared; this is due to the removal of one feature from the previous setting. However, the residual plots (see Figure 14) show the nonlinear relationship between the independent variables and dependent variables. The new features were introduced by applying nonlinear functions to the base features. This was tested with setting 5 and above.

Settings 3 and 4 did not yield any better performance values than settings 1 and 2. However, setting 5 showed a significant improvement in performance with a lesser p value and higher R -squared and adjusted R -squared values; this again confirms that the features (independent variables) are nonlinearly 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 a more significant p value and a small degree of freedom. The negative value of adjusted R -squared reveals that the

model suffers 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 settings, setting 9 with the triple feature crossing using stepwise AIC resulted in better results.

Stepwise linear regression feature selection based on setting 6 resulted in graphs; as shown in the figure, Figure 15(a) shows 10-fold crossvalidation results and Figure 15(b) shows leave-one-out (i.e., $k = n$) crossvalidation. Both graphs show that the best tune based on RMSE is when using 4 variables. Since the result is purely based on RMSE, it was not considered the best model.

The regression model on setting 8 is more prominent than the other seven models with a smaller p value and decent R -squared.

The model from setting 9 can be considered a proof of concept even though the model is complex to interpret

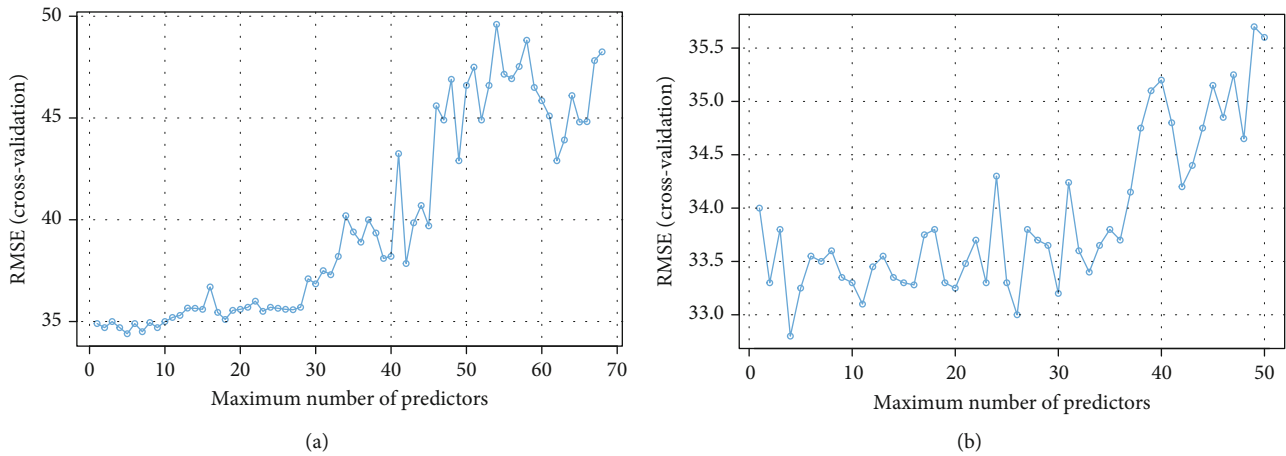


FIGURE 15: Stepwise regression feature selection using (a) 10-fold crossvalidation and (b) leave-one-out crossvalidation.

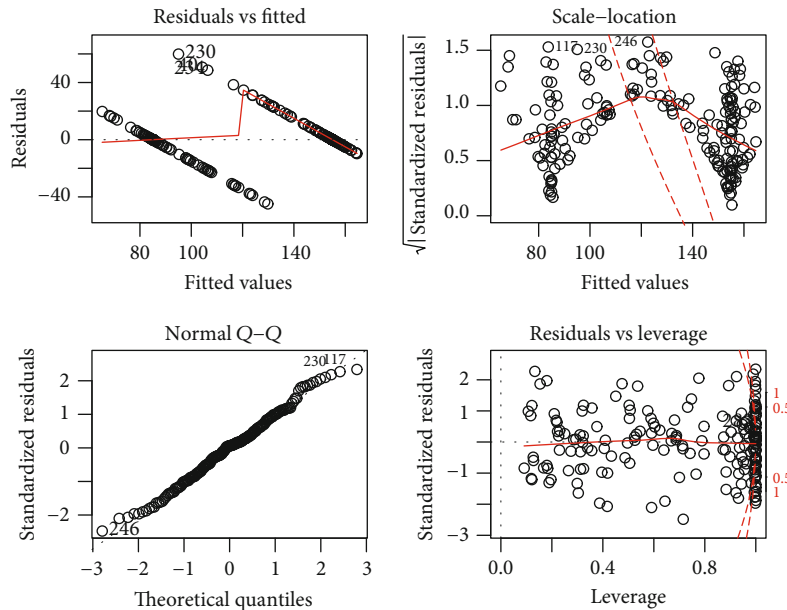


FIGURE 16: Setting 8 model plots.

and has the adjusted R -squared below 0.8. The model on setting 9 showed a better result with a very small p value, an elevated adjusted R -squared value, and a smaller standard residual error.

Figures 16 and 17 show 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 nonlinear 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 18 shows the error distribution of the inference testing using regression on setting 9, and it is safe to say that

the regression inference predicts ± 21 kg for 65% of the data, which is of $\pm 19\%$ accuracy in average for 65% cases, $\pm 38\%$ accurate with 95% confidence.

4.1. Model Performance. 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. In this paper, we have discussed the multiple nonlinear regression models, which have shown better performance for a smaller dataset. The result of this research shows strong evidence of the ability to infer the vehicle weight using VT data. Results reveal that a significant level of prediction could

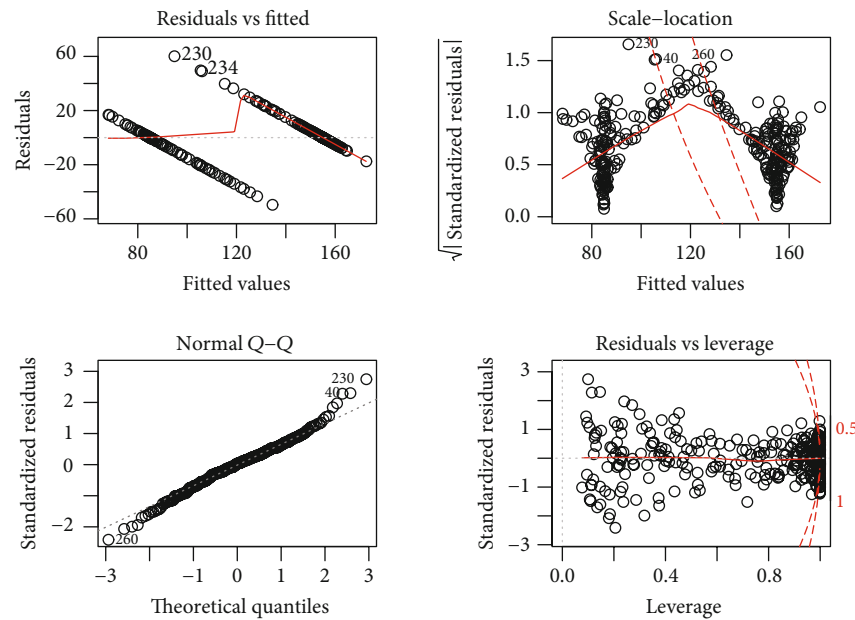


FIGURE 17: Model plots for setting 9.

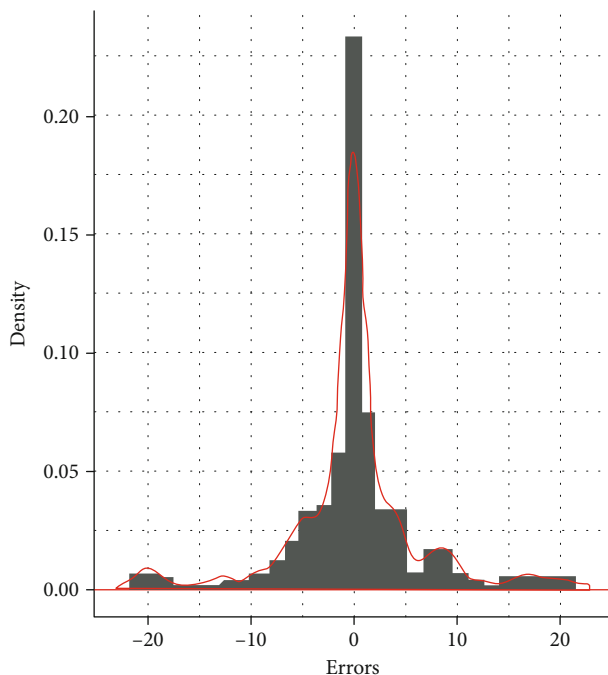


FIGURE 18: Distribution of errors for regression using setting 9.

be made using the selected features. The selected model has performed well, even with a small dataset. This is encouraging because in the real world, we cannot ask the vehicle owners to drive the vehicle several times with several different weights.

4.2. WIM System Performance. The performance of a WIM system is discussed by looking at many different factors. In here, we compare the proposed WIM system with other WIM systems using categorical values.

Table 8 discusses the performance comparison of the prototype WIM system with the existing WIN systems based on [6, 44–48]. The cost column in Table 8 compares the WIM systems based on installation, maintenance expenses, and labour cost. In comparison with other WIM solutions, the systems built using our approach would not have any maintenance cost or labour cost. Additionally, the installation cost could be negligible if the existing VT devices were 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.

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 [49]. But our proposed WIM system approach could be simply deployed on any compatible vehicles with an OBD 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 is usually ~100 kg. 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 because 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.

TABLE 8: 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	High (restricted)	Medium	Low	Low	Stationary/low
LS-WIM	Stress sensors	High	Moderate (restricted)	Moderate (annual)	Moderate	Moderate	Moderate
HS-WIM	Piezoelectric cable	High	Low (restricted)	High	Moderate	High	High
	Lines quarts	High	Low (restricted)	High	Moderate	High	High
Proposed WIM	Telematics	Low	Low (unrestricted)	Moderate	High	Low	High

The calibration frequency is reported higher in HS-WIMs than in static and LS-WIMs. With the proposed system approach, once a vehicle is trained with VT data, the retraining can be done at any time. This retraining process can be considered a calibration in other WIMs. This can be done in case of repeated false inference. 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. WIM systems built using the proposed approach will virtually be available everywhere on any road segment.

According to the literature, the sensor material used in HS-WIM is more fragile and prone to failure. Since the proposed WIM system approach does not need any such sensors and relies on robust ECU data, it has a lesser chance of failure. Once the data is available on the back-end server, the inference speed is nearly instantaneous. This makes the prototype system much faster in measuring speed. The other most important advantage of our approach is that it is scalable. Tests on the prototype WIM system built using the proposed approach shows $\pm 19\%$ inference accuracy on average for 65% cases, $\pm 38\%$ accurate with 95% confidence. This is near to the most HS-WIM systems. But compared to other WIM solutions, systems built using our approach can be scalable and cost-effective. We can use the existing data collection devices by insurance (UBI or PAYD) schemes. This would reduce the cost of implementation on a large scale. Communication technologies such as LoRaWAN (long range wide area network) [50] allow us to build fast, reliable, cheaper communication systems.

4.3. Assumptions and Limitations. This research was done based on several assumptions and limitations. According to McKay et al. [12], tyre pressure influences the detection accuracy. The recommended tyre pressure was maintained, and the pressure fluctuation due to the atmospheric temperature change was neglected. The influence of the size and the shape of tyres (tyre profile) was not considered in this research.

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

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

5. Conclusions

In this paper, we discussed the prototype design and development of a new WIM system using VT and ML. A prototype WIM system was developed and used as a proof of concept. Design considerations and the solutions used were discussed. The prototype is tested using a small car's VT data sample. The results show that it is possible to infer a weight of a vehicle using its telematics data. Multiple linear regression with setting 9 performed better than the other settings with the smaller standard residuals of 23.1, degrees of freedom of 88, significant p value of $6.322e-08$, better R -squared of 0.87, and a decent adjusted R -squared of 0.56. The result shows that, in the 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. 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 on overloading vehicles. The comparison of the prototype WIM system with the other existing systems showed that the proposed system approach can produce a cheaper, scalable, omnipresent, online (24/7) solution.

The performance of the prototype system on different vehicle types and different road and weather conditions needs to be researched. Other ML approaches such as decision trees, Bayesian inferencing, and neural networks need to be applied, and their performance comparisons are to be done in future research in this area.

Data Availability

The VT data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

References

- [1] G. Meiring and H. Myburgh, "A review of intelligent driving style analysis systems and related artificial intelligence algorithms," *Sensors*, vol. 15, no. 12, pp. 30653–30682, 2015.
- [2] National Road Traffic Act, 1996 (Act no. 93 of 1996) -248, Presumptions, 1999.
- [3] V. C. Magana and M. Munoz-Organero, "Toward safer Highways: Predicting Driver Stress in Varying Conditions on Habitual Routes," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 69–76, 2017.
- [4] M. Oastler, *Vehicle weights explained |tare, kerb, GVM, payload and trailer figures - Car Advice |CarsGuide*, Cox Automotive, COX AUSTRALIA MEDIA SOLUTIONS PTY LTD, 2015, December 2017, <https://www.carsguide.com.au/car-advice/vehicle-weights-explained-tare-kerb-gvm-payload-and-trailer-figures-37482..>
- [5] J. Richardson, S. Jones, A. Brown, E. O', N. A. Brien, and D. Hajilizadeh, "On the use of bridge weigh-in-motion for overweight truck enforcement," *International Journal of Heavy Vehicle Systems*, vol. 21, no. 2, p. 83, 2014.
- [6] J. Gajda, P. Burnos, and R. Sroka, "Accuracy assessment of weigh-in-motion systems for vehicle's direct enforcement," *IEEE Intelligent Transportation Systems Magazine*, vol. 10, no. 1, pp. 88–94, 2018.
- [7] W. Weidner, F. W. G. Transchel, and R. Weidner, "Classification of scale-sensitive telematic observables for riskindividual pricing," *European Actuarial Journal*, vol. 6, no. 1, pp. 3–24, 2016.
- [8] S. Kirushnath and B. Kabaso, "Weigh-in-motion using machine learning and telematics," in *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)*, pp. 115–120, Kuching, Malaysia, July 2018.
- [9] J. Labarrere, "What is WIM and history," in *ISWIM - International Society for Weigh in Motion*, ISWIM International Society for Weigh in Motion, 2017, December 2017, <http://www.iswim.org/index.php?nm=2&nsm=1&lg=en>.
- [10] I. L. Al-Qadi, H. Wang, Y. Ouyang, K. Grimmelsman, and J. E. Purdy, *LTBP Program's Literature Review on Weigh-In-Motion Systems* (No. FHWA-HRT-16-024), Federal Highway Administration. Office of Infrastructure Research and Development, USA, 2016.
- [11] R. Shah, Y. Sharma, B. Mathew, V. Kateshiya, and J. Parmar, "Review paper on overloading effect," *International Journal of Advanced Scientific Research and Management*, vol. 1, no. 4, pp. 131–134, 2016.
- [12] T. R. McKay, C. Salvaggio, J. W. Faulring et al., "Passive detection of vehicle loading," in *Visual Information Processing and Communication III*, Burlingame, California, USA, 2012.
- [13] T. Ihiguro, C. N. Yamada, A. T. Murahashi, and T. C. H. Kato, "Vehicle weight estimating device," US 7039519 B2, 2006.
- [14] S. K. W. Paul Hilgeman and R. Vicente, "System for controlling acceleration of a load coupled to a gas turbine engine," US006035629A, 2000.
- [15] D. Braun, E. Reiter, and A. Siddharthan, "Creating textual driver feedback from telemetric data," in *Proceedings of the 15th European Workshop on Natural Language Generation (ENLG)*, pp. 156–165, Brighton, UK, 2015.
- [16] E. CSS, "OBD2 Explained - A Simple Intro (2019)," in *CSS Electronics*, CSS Electronics, 2018, January 2019, <https://www.csselectronics.com/screen/page/simple-intro-obd2-explained>.
- [17] R. Vaiana, T. Iuele, V. Astarita et al., "Driving behavior and traffic safety: an acceleration-based safety evaluation procedure for smartphones," *Modern Applied Science*, vol. 8, no. 1, pp. 88–96, 2014.
- [18] D. I. Tselentis, G. Yannis, and E. I. Vlahogianni, "Innovative motor insurance schemes: a review of current practices and emerging challenges," *Accident Analysis & Prevention*, vol. 98, pp. 139–148, 2017.
- [19] S. Kirushanth and B. Kabaso, "Telematics and Road Safety," in *2018 2nd International Conference on Telematics and Future Generation Networks (TAFGEN)*, pp. 103–108, Kuching, Malaysia, July 2018.
- [20] E. Massaro, C. Ahn, C. Ratti et al., "The car as an ambient sensing platform [point of view]," *Proceedings of the IEEE*, vol. 105, no. 1, pp. 3–7, 2017.
- [21] U. Kiencke, S. Dais, and M. Litschel, "Automotive serial controller area network," in *SAE Technical Paper Series*, p. 8, USA, February 1986.
- [22] Elmelectronics, *OBD to RS232 Interpreter ELM327*, Elm Electronics, 2011.
- [23] E. I. Vlahogianni and E. N. Barmounakis, "Driving analytics using smartphones: algorithms, comparisons and challenges," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 196–206, 2017.
- [24] D. Chen, K. T. Cho, S. Han, Z. Jin, and K. G. Shin, "Invisible sensing of vehicle steering with smartphones," in *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '15*, pp. 1–13, Florence, Italy, 2015.
- [25] M. Amarasinghe, S. Kottegoda, A. L. Arachchi, S. Muramudalige, H. M. N. Dilum Bandara, and A. Azeez, "Cloud-based driver monitoring and vehicle diagnostic with OBD2 telematics," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, pp. 505–510, Dekalb, IL, USA, May 2015.
- [26] J. E. Meseguer, C. T. Calafate, J. C. Cano, and P. Manzoni, "Drivingstyles: A smartphone application to assess driver behavior," in *2013 IEEE Symposium on Computers and Communications (ISCC)*, Split, Croatia, July 2013.
- [27] K. Takeda, C. Miyajima, T. Suzuki et al., "Self-coaching system based on recorded driving data: learning from one's experiences," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1821–1831, 2012.
- [28] J. Ferreira, E. Carvalho, B. V. Ferreira et al., "Driver behavior profiling: an investigation with different smartphone sensors and machine learning," *PLoS One*, vol. 12, no. 4, 2017.
- [29] J. T. Pohlmann and D. W. Leitner, "A comparison of ordinary least squares and logistic regression," *Ohio Journal of Science*, vol. 103, no. 5, 2003.
- [30] A. B. Chan and N. Vasconcelos, "Counting people with low-level features and Bayesian regression," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 2160–2177, 2012.
- [31] A. Alessandrini, F. Filippi, and F. Ortenzi, "Consumption calculation of vehicles using OBD data," in *2012 International Emission Inventory Conference "Emission Inventories - Meeting the Challenges Posed by Emerging Global, National, Regional and Local Air Quality Issues"*, pp. 1–21, Tampa, FL, USA, 2012.
- [32] S.-C. Lin and K.-C. Li, "Systems and methods for vehicle load detection and response," US20160355189 A1, 2016.

- [33] International Organization for Standardization, *ISO 15031 - road vehicles - communication between vehicle and external equipment for emissions-related diagnostics*, International Organization for Standardization, 2015.
- [34] S. Kirushanth and B. Kabaso, "Designing a cloud-native weigh-in-motion," in *2019 Open Innovations (OI)*, pp. 25–29, Cape Town, South Africa, October 2019.
- [35] J. Hurwitz and D. Kirsch, *Machine Learning for Dummies IBM Limited Edition*, John Wiley & Sons, Inc., 2018.
- [36] A. González, A. T. Papagiannakis, and E. J. O'Brien, "Evaluation of an artificial neural network technique applied to multiple-sensor weigh-in-motion systems," *Transportation Research Record*, vol. 1855, no. 1, pp. 151–159, 2003.
- [37] D. H. Oh, *Cheat Sheet : Kubernetes*, JBoss Community, 2018.
- [38] T. Gluster, *Architecture - Gluster Docs*, Red Hat, 2019, May 2019, <https://docs.gluster.org/en/latest/Quick-Start-Guide/Architecture/>.
- [39] P. R. Wieringa, *Design Science Research in Information Systems and Software Systems Engineering Research Methodology Accross the Disciplines*, Università della Svizzera italiana, 2016.
- [40] T. Swinscow, "Statistics at Square One: 11. Correlation and Regression," *BMJ*, vol. 16, no. 22, 1997.
- [41] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, no. 10, p. 78, 2012.
- [42] A. Kassambara, *Stepwise regression essentials in R - articles - STHDA*, STHDA, 2018, February 2019, <http://www.sthda.com/english/articles/37-model-selection-essentials-in-r/154-stepwise-regression-essentials-in-r/>.
- [43] R. R. Hocking, "A biometrics invited paper. The analysis and selection of variables in linear regression," *Biometrics*, vol. 32, no. 1, p. 1, 1976.
- [44] B. Timerson, *TECHNICAL Evaluating Weigh-in-Motion Systems for Accuracy and Durability*, Minnesota Department of Transportation, 2018.
- [45] B. Jacob and V. F. de La Beaumelle, "Improving truck safety: potential of weigh-in-motion technology," *IATSS Research*, vol. 34, no. 1, pp. 9–15, 2010.
- [46] M. Lydon, S. E. Taylor, D. Robinson, A. Mufti, and E. J. O'Brien, "Recent developments in bridge weigh in motion (B-WIM)," *Journal of Civil Structural Health Monitoring*, vol. 6, no. 1, pp. 69–81, 2016.
- [47] M. R. Karim, N. I. Ibrahim, A. A. Saifizul, and H. Yamanaka, "Effectiveness of vehicle weight enforcement in a developing country using weigh-in-motion sorting system considering vehicle by-pass and enforcement capability," *IATSS Research*, vol. 37, no. 2, pp. 124–129, 2014.
- [48] US Department of Transportation, *Weigh-In-Motion Pocket Guide Part 2 WIM Site Selection, Design, and Installation Guide*, US Department of Transportation, 2018.
- [49] T. Haugen, J. R. Levy, E. Aakre, and M. E. P. Tello, "Weigh-in-Motion Equipment - Experiences and Challenges," *Transportation Research Procedia*, vol. 14, no. 2352, pp. 1423–1432, 2016.
- [50] L. R. Alliance, *About LoRaWAN® | LoRa Alliance™*, LoRa Alliance, 2019, April 2019, <https://loro-alliance.org/about-lorawan..>