

# Real-time mass estimation of connected commercial vehicle using artificial neural network model

Chandrasekaran M\* & Rajesh PK

Department of Automobile Engineering, PSG College of Technology, Avinashi Road, Peelamedu, Coimbatore 641 004, Tamil Nadu, India

Received: 22 January 2024; Accepted: 10 May 2024

In recent years, the automotive industry has experienced a remarkable transformation with the advent of digital technologies. The internet of things (IoT) revolutionizes the automobile industry by enabling intelligent, connected vehicles. IoT-generated data enhances vehicle safety through real-time diagnostics, predictive maintenance, and remote monitoring, reducing accidents and breakdowns. Very few studies have used IoT data thread inference in product design. Knowing the real-time mass of the vehicle is significant for design engineers during aggregate design selection and optimizing the vehicle design. This study shows a novel approach to predicting the real-time active mass of a connected medium-duty commercial truck using an artificial neural network (ANN) deep learning (DL) multilayer perceptron (MLP) deep learning algorithm. In this process, the raw data collected from the vehicle is cleaned, and the vehicle's mass is estimated by applying the vehicle dynamics system longitudinal forces model. Different load conditions of the vehicle are calculated with an accuracy of 87%. Later, the estimated mass with the five mass-influencing operating parameters from the data is used as an input in the MLP deep learning model to predict the vehicle's mass as output. The model is trained and tested using overload, rated load, and no-load conditions; when testing the model using the real-time operating parameters, the deep learning model predicted the mass with >90% accuracy. This deep learning model, when integrated into the data-driven digital twin framework, will be instrumental in controlling various actuators based on the predicted mass in future work. Moreover, the predicted real-time active mass is not only helpful for the optimum design of many vehicle systems but also for building application-based design configurations, thereby demonstrating the practical relevance and potential applications of this research in vehicle design and control systems.

**Keywords:** Data-driven vehicle design, Longitudinal vehicle dynamics model, Mass prediction, Medium-duty truck, Multilayer perceptron algorithm

## 1 Introduction

In commercial vehicle design, the vehicle's mass plays a significant role in transportation, logistics, and engineering. Accurate mass estimation is crucial for various purposes, such as safety, design optimization, fuel efficiency, and regulation compliance. The active mass is used in designing vehicle control systems such as economic cruise control<sup>1</sup>, automated gear shift<sup>2</sup>, strategies for optimization of fuel consumption<sup>3-4</sup>, cruise control<sup>5</sup>, control of anti-lock braking systems<sup>6</sup>, regenerative braking control strategy<sup>7</sup>, electric park brake and power management strategy<sup>8</sup>, vehicle stability and control<sup>9</sup> and other control systems. Numerous studies demonstrate the significance of predicting vehicle mass and road grade<sup>10-11</sup>. Vehicle mass plays a significant role in electric vehicle (EV) design in predicting the energy consumption of battery discharge<sup>12</sup> and the distance

coverage<sup>13</sup>. As a vehicle management criterion, it is evident that vehicle mass is critical to the intelligent operation of the vehicle<sup>14</sup>. Thus, finding the accurate mass of the vehicle becomes a more significant and exciting topic for automotive researchers.

### 1.1 Mass estimation approach

In the past, the vehicle mass was estimated using three significant approaches: sensor-based approach, model-based approach, and machine learning approach.

#### 1.1.1 Sensor-based approach

In the sensor-based mass estimation, the vehicle is mounted with multiple sensors. Load cells are sensors that can be installed in various parts of a vehicle to measure the weight applied to those specific areas. You can estimate the total vehicle mass by summing up the measurements from these sensors. Axle load sensors are placed under each vehicle's axle to measure the load on each axle. You can estimate the

\*Corresponding author (E-mail: chandrasekaran.me@gmail.com)

total vehicle mass by combining the axle load measurements. Pressure sensors can be used to estimate the mass of a vehicle by measuring the pressure of the vehicle's tires. This method is less accurate than others and is often used for rough estimates. Ultrasonic sensors can be mounted on a vehicle's suspension system to measure the distance between the vehicle and the road surface. Using this data, you can estimate the vehicle's weight based on its suspension behaviour. Strain gauges can be attached to critical structural components of a vehicle, such as the chassis or suspension, to measure the deformation caused by the vehicle's weight. From this data, the mass can be estimated. Global positioning system (GPS)<sup>10,15-17</sup>, and displacement transducer<sup>18</sup>, in which the road grade is calculated initially, and then the vehicle mass is estimated. Moreover, GPS signals may only be available sometimes for part of the trip, and there may be data loss during the intermediate trip of the vehicle, which may affect the accuracy of estimating the mass. This approach provides better accuracy in estimating the mass. However, fitting additional sensors and measuring the vehicle's mass is too expensive, and the sensor's maintenance and calibration add up to the operational cost of this approach.

### 1.1.2 Model-based approach

Thus, the need for an estimation model to calculate the mass with limited sensor data passed into a vehicle dynamic model evolved. While using the longitudinal dynamic model, there is a coupling of road slope and vehicle mass; thus, Zhang et al.<sup>19</sup> used recursive least squares (RLS) with a forgetting factor to estimate the road slope and mass of the vehicle instantaneously. Cai et al.<sup>20</sup> proposed a two-layer structure algorithm, which discusses road slope estimation using RLS and vehicle mass estimation based on an extended Kalman filter (EKF). Seungki et al.<sup>21</sup> used the property of sensor signals of vehicle acceleration that contained road slope information to estimate the road slope and vehicle mass. Zhao et al.<sup>22</sup> applied fuzzy logic to longitudinal speed, longitudinal acceleration, and engine torque data to calculate the vehicle mass confidence factor. When the confidence factor exceeded a specified limit, the vehicle mass was approximated using RLS with the forgetting factor. Sun et al.<sup>23</sup> utilized EKF to estimate the road slope and then used EKF and RLS to estimate the vehicle mass based on the predicted road slope. The final estimate of the vehicle's mass was generated

from a weighted average of estimations from two approaches with constant weights. Studies have been on applying frequency domain analysis to estimate vehicle mass. Hu et al.<sup>24</sup> developed the longitudinal acceleration and wheel angular speed amplitude-frequency functions by the longitudinal vehicle dynamic model, then used the least square approach to estimate vehicle mass; nevertheless, the effect of real automobile tests needs additional investigation. Chu et al.<sup>25</sup> used high-frequency information extraction and the RLS technique to decouple vehicle mass and road slope and estimate vehicle mass. Dynamics-based techniques, in general, exhibit rapid convergence and excellent stability. However, proper vehicle dynamic model parameters and state parameter monitoring are necessary for the effectiveness of dynamics-based techniques. Furthermore, while dynamics-based techniques may produce accurate vehicle mass estimation findings when the vehicle is subjected to significant and persistent excitation, the estimation accuracy is difficult to guarantee when the vehicle is subjected to minor excitation. Kenneth et al.<sup>30</sup> proposed a longitudinal dynamic model to estimate ground vehicles' real-time mass based on the longitudinal dynamics model. The mass is estimated using controlled area network (CAN) bus data, and the 6-axis inertial measurement unit (IMU) is processed using least square filters with forgetting factor and validated the model using the data from two different petrol-powered passenger cars. The resulting mass estimates were reasonable, with +/- 5% variations with actual mass. However, the primary factors influencing the tuning of the mass model were the rolling resistance coefficient and the mass moments of inertia of the engine and wheels. The parameters, however, are consistently maintained at a constant level while comparing various vehicle loads to ensure the integrity and reliability of the obtained outcomes. S. Mutta et al.<sup>33</sup> compared the EKF and RLS estimation algorithms to estimate the road slope and vehicle mass using model-based algorithms. Drive-train torque, vehicle velocity, and vehicle design parameters are inputs for the model. The model proved that the EKF filter is better than the RLS filter. Still, the accuracy of the longitudinal dynamics in low gears needs to be improved, and the vehicle accessory load adaption affected the accuracy. Wang B et al.<sup>34</sup> estimated the mass of the truck based on onboard sensors. This method analyzed the factors that affect the mass estimation, and the error correction method

was introduced using the sensor data. Later, the mass estimation algorithm was established to predict the vehicle's mass. The experiment results were encouraging and provided a 1.1% average relative error. Cai L et al.<sup>35</sup> estimated the mass of the commercial vehicle using a two-layer structure algorithm. The first layer estimated the road grade using the recursive least squares method and passed the grade to the second layer, which evaluated the mass using an extended Kalman filter. The sensors were fitted in the rear suspension to measure the suspension deformation on the grade model. The estimated algorithm was validated using MATLAB/Simulink and the TruckSim platform. The root mean square error (RMSE) of these models drops from 871.65 to 772.52, encouraging road grade estimation and vehicle mass estimation by the EKF method. Lin N et al.<sup>36</sup> propose estimating vehicle mass considering system error in the vehicle longitudinal dynamics model. Vehicle mass and system error were considered two unknown parameters, and the RLS algorithm was used to estimate the mass. The accuracy of vehicle mass improves as the system error decreases.

### 1.1.3 Machine learning approach

Machine learning (ML) has been extensively used in intelligent driving, predicting vehicle speed using the recurrent neural network (RNN) algorithm<sup>26</sup> and predicting fuel consumption using the artificial neural network (ANN) algorithm<sup>27</sup> in recent years. However, ML has yet to be exploited in vehicle mass estimation research. The proposed approach in this paper synthesizes the dynamics model and ANN deep learning machine learning model. The literature below encourages using deep learning machine learning models for mass estimation. Torabi et al.<sup>28</sup>, using acceleration, engine torque, vehicle velocity, and time as inputs, proposed a feed forward neural network (FFNN) technique for calculating highway road grade and vehicle mass for heavy-duty vehicles. Even though this approach has been more accurate, the average RMS error for road grade and heavy-duty mass estimation was 0.10-0.14 degrees and 1%, respectively. The algorithm cannot sustain the accuracy when the road slope's magnitude is significant, or the vehicle acceleration or torque varies rapidly. Korayem et al.<sup>29</sup> investigated both a dynamic system model-based technique and a machine learning approach to estimating trailer mass using a feed forward neural network approach with 15 fully connected layers to calculate trailer mass, which

demonstrated a more vital ability to estimate mass appropriately during braking phases of driving, which model-based techniques may find challenging to achieve. Yu. Z. et al.<sup>31</sup> used a similar approach using the FFNN algorithm to predict the operating parameters. They passed the parameters to the dynamic model-derived RLS with a forgetting factor to estimate the mass. In this approach, the accuracy was good in the 80-90% range for the new Europe driving cycles (NEDC), a controlled environment test for passenger cars and not practically tested with commercial vehicles. M. Eagon et al.<sup>32</sup> estimated the mass of heavy-duty vehicles using Gaussian belief propagation (GBP). In this approach, the k-nearest factors (kNF) and combining their adjacent beliefs proposed the kNF-GBP mass prediction model, which performed better than the k-nearest neighbor (KNN) regressor. Both kNN and kNF-GBP mass prediction models predicted the mass with coefficients of determination above 0.97, which was encouraging.

### 1.2 Novel approach

Table 1 above compares various works of literature on mass estimation. The authors have used a two-step methodology of combining vehicle dynamics and machine learning models in all these works. However, in all cases, the vehicle dynamics model is used as a mathematical model to simulate the mass data for various simulated driving and load conditions. The simulated data was used to train and test the machine-learning model. The FFNN algorithm used in all this literature, while feed forward neural networks can be powerful tools for regression analysis, they require careful consideration of various factors, including data preprocessing, model architecture, hyper parameter tuning, and handling of common challenges like over fitting and interpretability. The proposed model in this research work is trained with the real-time operational parameters of the vehicle and uses a classic multilayer perceptron deep learning model. MLPs can learn complex, non-linear relationships between input features and the target variable. This flexibility allows them to capture intricate patterns in the data, making them suitable for regression tasks where linear models may fall short. MLPs can be used for various regression tasks, including single-output and multi-output regression. You can modify the output layer to suit the specific requirements of your regression problem, such as predicting multiple continuous variables simultaneously. Many well-established deep learning libraries provide easy-to-use

Table 1 — Comparison of various literature using Machine Learning model for mass estimation

Sl. No	Author	Model used	Calculated	Observation	Conclusion	Year
1	Torabi et al. [28]	FeedForward Neural Network (FFNN) technique	Road grade and Vehicle mass for heavy-duty vehicles	(RMS) error for road grade and heavy-duty mass estimation was 0.10-0.14 degrees and 1%, respectively.	The algorithm cannot sustain the accuracy when the road slope's magnitude is significant, or the vehicle acceleration or torque varies rapidly.	2019
2	Korayem et al. [29]	Dynamic system model-based technique and Machine Learning FFNN approach	Trailer mass of the vehicle	Ability to estimate mass appropriately during braking phases of driving	The results justify the accuracy of the mass of the trailer in various cases can be estimated with less than 10% error.	2020
3	Yu. Z. et al. [31]	FeedForward Neural Network (FFNN) technique and Dynamics based method recursive least square (RLS) with forgetting factor based on vehicle dynamic model	Mass of the vehicle	Estimated the mass using two FFNN and RLS methods and fused it into fuzzy logic. The accuracy was good in the 80-90% range for the New European Driving Cycles (NEDC) condition	Controlled environment tests are used for passenger cars and are not practically tested with commercial vehicles.	2022
4	M. Eagon et al. [32]	Gaussian Belief propagation (GBP)	Mass of the vehicle	Estimating the mass of road vehicles by borrowing concepts from kNN to improve the robustness of GBP-based probabilistic inference	Both kNN and kNF-GBP, mass prediction models, predicted the mass with coefficients of determination above 0.97 was encouraging.	2022
5		Research Paper	Real-time mass of the medium-duty truck	Predicting the real-time mass of the medium-duty truck by training the MLP ANN algorithm using vehicle longitudinal dynamics estimated mass	In this method, the accuracy of the model is >90% with an RMSE of 0.003	2023

tools for building, training, and evaluating MLP models. This makes implementing and experimenting with MLPs for regression tasks relatively straight forward. The novelty of the proposed model is explained below.

**Self-sustainable model:** This approach's novelty is its two-step methodology. The first step identifies the relevant features from the IoT data threads used for estimating the mass; this step helps calculate the vehicle's real-time mass using the real-time operating parameters/features and allows the model to be self-sustainable.

**Generic system model:** This dynamics system model is designed as a standard system model for alltrucks with varying gross vehicle weight (GVW), and segments of vehicles like GVW range from 8,000 Kgs to 49,000 Kgs, and the truck segments like haulage, tipper, and tractors. The model of the vehicle is one of the input parameters for the ANN model, and the trained model is used to predict the results.

**Retrain the MLP Model:** The real-time data will be used to predict the vehicle's mass using the MLP model. If the predicted mass accuracy deteriorates, the system model can generate mass for the current set of data points and use the estimated mass to retrain the MLP model and improve the accuracy.

## 2 Materials and Methods

The objective is to predict the operating load of the vehicle with a deep learning artificial neural network (ANN) algorithm using the real-time operational parameters of the vehicle collected using internet of things (IoT) data threads. This is done using a two-step approach, as shown in Fig. 1. In step 1, the system model is developed using vehicle longitudinal dynamics forces. Raw data collected from the vehicle sensors are cleaned and passed into the model to get the estimated mass, which is validated using a z-score formula to get the best-fit mass from the calculated mass.

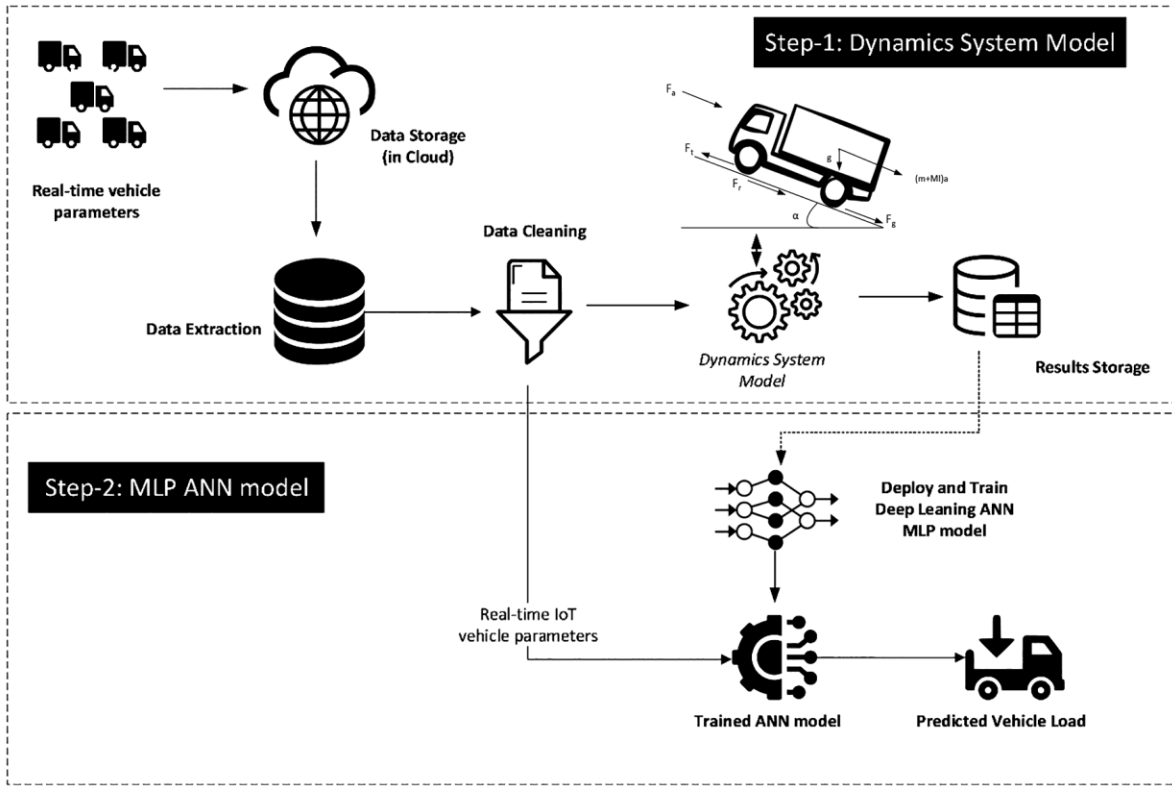


Fig. 1 — Schematic diagram of the methodology of the model.

In step 2, the estimated mass for every trip with the operating parameters is used to train and test the ANNMLP deep learning model, which is later used to predict the real-time mass of the vehicle. In this paper, the methodology discusses the medium commercial vehicle (MCV) haulage vehicle segment. The model used in this experiment was a 4x2 truck with a gross vehicle weight of 19 tonnes or 19,000 Kg fitted with a 6-cylinder diesel engine that generates a max power of 200 hp. In this research paper, the data used and discussed was with this model only. The technical specifications for the vehicle are given in Table 2. 20,000+ vehicles are running in the field, and the real-time data from all these vehicles are stored in a cloud called “Big Data” in this paper, which is also the “Raw data.”

**2.1 System model using vehicle longitudinal dynamics**

The mass estimation is based on the vehicle longitudinal dynamics model, which describes the relationship between the tractive force and driving resistances. The free-body diagram of the longitudinal motion analysis of the commercial vehicle is shown in Fig. 2. Conferring to Newton’s second law, the longitudinal dynamic model of a vehicle can be referred to as,

Cabin	Fully built cabin
Engine	6-cylinder Euro VI engine
Max power	200 HP @ 2400 rpm
Max torque	700 Nm @ 1200 – 2000 rpm
Clutch	380 mm dia diaphragm with clutch booster
Gearbox type	6-speed synchromesh
Rear axle	Fully floating single-speed rear axle, hypoid
Front axle	Forged I section – Reverse Elliot type
Steering	Integral tilt and telescopic power steering
Front Suspension	Semi-elliptic leaf spring with shock absorbers
Rear Suspension	Semi-elliptic multi-leaf
Chassis frame	HSS material with new, improved design
Brakes	Full air dual line brakes
Fuel tank	185L / 350L
AdBlue tank	24L
Electricals	24 V, 110 AH, alternator – 85A
Tyres	295/90R20 tube tyres
Loading spans	5.5 m/ 6.1 m/ 7.3 m/ 9.3 m
GVW	18,500 kg

$$F = m \times a \quad \dots (1)$$

Where F is force, m is mass, and a is acceleration. Applying this equation (1) with the practical forces as per vehicle longitudinal dynamics can be rewritten as,

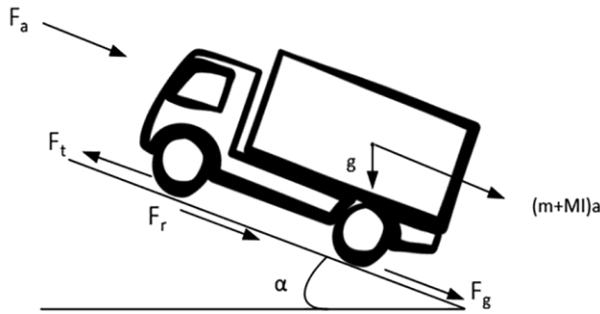


Fig. 2 —Schematic diagram of vehicle longitudinal dynamics.

$$M \cdot a = F_t - F_r - F_g - F_a \quad \dots (2)$$

The forces acting on the moving vehicle are calculated by reducing the total resistance formed during the vehicle’s movement from the tractive force generated for moving the vehicle.  $F_t$  is the Tractive force, in N,  $F_r$  is the rolling resistance, in N,  $F_g$  is the gradient resistance, and in N,  $F_a$  is the air resistance in N. In a moving vehicle, the mass is increased due to the inertia of the rotating components; thus, the force that is derived contains the mass of the vehicle, and the mass of the rotating components is generated due to the moment of inertia. Thus, the mass in the equation is represented as the sum of mass and moment of inertia of the vehicle caused during the vehicle’s movement, as shown in Equation (2). The mass addition created by a moment of inertia of the rotating components in the vehicle is calculated by considering the engine inertia and wheel inertia, which are the primary mass impacting rotating components in the vehicle. The Equation (3) given below is used to calculate the moment of inertia of the rotation components of the vehicle.

$$M = \left[ m + \frac{e_i(i_g i_r)^2 + W_i}{r^2} \right] \quad \dots (3)$$

$M$  is the total mass,  $m$  is the mass of the vehicle,  $e_i$  is engine inertia,  $\text{kgm}^2$ ,  $i_g$  is gear ratio, dimensionless,  $i_r$  is rear axle ratio, dimensionless,  $W_i$  is wheel inertia,  $\text{kgm}^2$  and  $r$  is wheel radius, in meters (m).  $R$  is the overall conversion factor from engine torque to force at the wheels, based on the outer tire radius  $r$  and the transmission and differential gear ratios,  $i_g$  and  $i_r$ , respectively; thus, equation (3) is simplified as Equation (5).

$$R = \frac{i_g i_r}{r} \quad \dots (4)$$

$$M = \left[ m + e_i(R(t))^2 + \frac{W_i}{r^2} \right] \quad \dots (5)$$

In this Equation (5), engine inertia, wheel inertia, and wheel radius are constants for each vehicle segment. The overall conversion factor Equation (4) has variables that are calculated based on the telematics data, such as Wheelbased speed and engine speed, which plays a vital role in determining the moment of inertia of the rotating components like engine and wheel, the overall conversion factor which contains the same is the only variable which differs on telematics data thus mentioned as  $R(t)$ . When applying Equation (5) in Equation (2), the below Equation (6) is derived,

$$\left[ m + e_i(R(t))^2 + \frac{W_i}{r^2} \right] a = F_t - F_r - F_g - F_a \quad \dots (6)$$

The traction force is generated to move the vehicle on the surface by overcoming the friction or resistance caused between a truck and the road or tangential force. To simplify and formulate the empirical formula for tractive force is generated by the engine, which produces the energy to move the vehicle and the overall ratio of the driveline to move the wheel of the vehicle as shown in Equation (6)

$$\text{Tractive Force, } F_t = T_e(t)R(t)\eta_d \quad \dots (7)$$

$T_e$  is engine torque, Nm, and  $\eta_d$  is driveline efficiency, % of the vehicle. In Equation (7), the telematics data points like torque, frictional torque, and engine speed are used to derive the engine torque thus  $T_e(t)$ .

The *rolling resistance* can be expressed by the generic Equation (8), as shown below,

$$F_r = C_r \cdot m \cdot g \cos(\alpha(t)) \quad \dots (8)$$

$C_r$  is the rolling resistance coefficient, dimensionless,  $m$  is the mass of the body, kg,  $g$  is the acceleration of gravity (9.81 m/s<sup>2</sup>), and  $\alpha$  is the road angle, radians.

The most basic type of resistance is grade resistance. The gravitational pull is at work on the vehicle. This force may not be precisely perpendicular to the roadway surface, especially when there is a gradient.

$$F_g = m \cdot g \cdot \sin(\alpha(t)) \quad \dots (9)$$

$m$  is the mass of the vehicle, kg,  $g$  is acceleration constant, m/s<sup>2</sup>,  $\alpha$  is road angle, radians.

$$\text{Altitude grade, } \alpha = \frac{\text{Altitude}_{i+1} - \text{Altitude}_i}{\text{GPS Haversine distance} * 1000} \dots (10)$$

In Equations (8) and (9), the road angle is the variable that is calculated using the haversine distance Equation (10) using the GPS data from the telematics device, by passing altitude, latitude, and longitude data thread of the trip is primarily used to calculate the resistance by measuring the angle of the road, as that is the dynamic data point used to know the angle of the road based on which the resistance modifies. The derivation of haversine distance based on GPS information<sup>37-38</sup> is beyond the scope of this paper.

When a body moves, the air around it produces resistance in the opposite direction of the movement, referred to as air resistance, expressed in Equation (11).

$$F_a = 0.5 \rho \cdot C_d \cdot C_a \cdot A \cdot v(t)^2 \dots (11)$$

$F_a$  is air resistance,  $\rho$  is air density,  $C_d$  is drag coefficient,  $C_a$  is the altitude correction factor,  $A$  is frontal area, and  $v$  is vehicle speed. In equation (11), the density of air is constant as it is not practical to measure the density of the air consistently. The coefficient of drag, altitude correction factor, and frontal area of the cabin, which resists air drag, are the constants for each vehicle model. The vehicle speed is required to calculate the drag force generated from the vehicle, which is fed from the Wheel based speed telematics data thread from the telematics big data.

Acceleration is the rate of change in an object's velocity over time. Equation (12) provides the general acceleration formula.

$$\text{Acceleration, } a = \Delta v / \Delta t \dots (12)$$

The derived acceleration formula for the analysis is given below in Equation (12), where the speed of the moving vehicle is taken from the telematics big data thread, and the sampling time is measured every 6 seconds.

$$\text{Acceleration, } a = \frac{(\text{Difference in Wheel based speed}) * 5}{18 * \text{sampling time in seconds}} \dots (13)$$

All these derivatives are applied to the mass formula, as shown in equation (6), simplified as Equation (14). This will be the base equation for estimating the vehicle's mass using real-time telematics data threads.

$$m(t) = \left[ \frac{T_e(t)R(t)\eta_d - \frac{1}{2}\rho \cdot C_d \cdot C_a \cdot A \cdot v(t)^2 - (e_i(R(t)))^2 + \frac{W_i}{r^2}a(t)}{a(t) + C_r \cdot g \cos(\alpha(t)) + g \cdot \sin(\alpha(t))} \right] \dots (14)$$

Equation (14) involves four significant components: tractive force, total resistance, acceleration, and moment of inertia. The feature mapping from the telematics big data threads needs to be mapped to respective variables in the equation to get the estimated mass of the vehicle. These data are time-bound for every 1 second. The data packets from the vehicle are collected and passed into the above variable components to estimate the mass. Table 3 clearly shows the mapping of formula components, the governing variables, and the equivalent vehicle feature mapping. These 8 data threads torque, frictional torque, engine speed, altitude, latitude, longitude, wheelbase speed, and sampling time from the big data are filtered, cleaned, and passed into Equation (14) to estimate the final mass of the vehicle. The mass estimation process is programmed in R language to speed it up. Thus, five IoT data threads will be used for the estimation of mass: vehicle model (M), Engine Torque (Q), frictional torque (F), engine speed (E), and Altitude grade (D). This is derived using altitude, latitude, longitude data, and wheelbase speed (W). These data points are extracted from the IoT big data and passed to the Equation (14) to get the estimated mass. Thus, these parameters are called mass-influencing parameters.

## 2.2 Validation of System Model

### 2.2.1 Data collection and storage

The primary control units that are used to control the operating parameters of the vehicle are the engine

Table 3 — Feature mapping against the formula component of the vehicle longitudinal dynamics model

Formula components	Governed by	Feature mapping
Tractive Force	Engine Torque, Overall Efficiency, Radius of Wheel	Torque, Frictional Torque, Engine Speed
Total Resistance	Air Resistance, Grade Resistance, Rolling Resistance	Altitude, Latitude, Longitude Wheelbased speed
Acceleration	Speed of the vehicle and time	Wheelbased speed, sampling time
Moment of Inertia	Engine Inertia, Wheel Inertia	Wheelbased speed, Engine speed

control unit (ECU), anti-lock braking system (ABS), onboard diagnostics (OBD), and after-treatment control unit (ACU). The data collected from these control units are aggregated in the telematics device; 110 real-time operating parameters are gathered at a frequency of 1 Hz or 1 second. The data from various control units are managed using the CAN data bus connectors. Transfer control protocol (TCP), unified diagnostic services (UDS), and message queuing telemetry transport (MQTT) protocols transfer the data from the control units to the telematics device. This telematics device, which acts as an edge device, also collects GPS data from the vehicle tracking system. The collected data is converted into data packets by identifying a unique ID, and the data is sent to the cloud server using a predefined authentication mechanism programmed in the controller. The data is transmitted from the control units to the cloud servers using the onboard telematics device using the embedded subscriber identity module (e-SIM) with a unique number activated through a network service provider. The data transfer authentication is validated from the centralized control unit to start data transfer to the cloud server. The data transfer rate is also predefined.

2.2.2 Extraction of data

The accuracy of the output depends on the quality of the input data. Thus, the process of cleaning the data is highly crucial. Figure 3 below shows the different layers of big data that need to be cleaned to get the feature data for analysis. The data cleaning process is done using three steps, they are,

Step 1: Extraction of vehicle data

First, the vehicle details need to be extracted from the big data. The vehicle identification number (VIN)

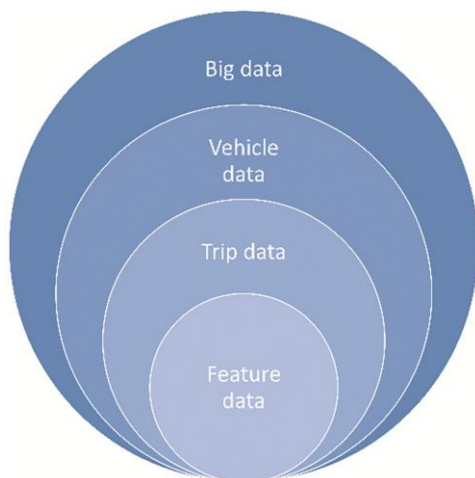


Fig. 3 — Layers of big data for analysis.

is used to identify the vehicle model and engine type, and the data packets are grouped based on the vehicle model. Figure 3 below explains the logic for extracting the GVW, engine type, and vehicle type from the VIN based on the number's 4th, 5th, and 7th digits and grouping the data packets based on the vehicle model.

Step 2: Extraction of trip data

The vehicle data is segregated from the above Step-1 logic and contains the data for many trips run in a day. This data needs to be cleaned to extract the data packets for a trip. The vehicle could have run for single or multiple trips with different masses in a day. Thus, breaking the data packets into single or multiple trips is essential. The trip separation logic breaks the vehicle data into trip(s) data. This logic doesn't need to be explained in detail as this is less relevant to this approach.

Step 3: Feature data extraction

After splitting the trip data, each trip data is provided with a unique ID with a sequence number based on the timestamp. From this trip data, only the necessary data threads are filtered and used as feature data for mass estimation.

2.2.3 Driving data filters

The extracted feature data from the filter contains extensive data. As this is real-time driving data, the challenge is eliminating unnecessary data points, which could be more beneficial for estimation. During real-time driving, the vehicle will be halted in signal, traffic, etc. During that time, the idling vehicle parameters will be captured, which is not applicable; also, some outlier data points will disturb the estimation. Thus, a few data filters are meticulously applied to the data to ensure the precision of the analysis. There are two types of data filters, as shown in Fig. 4, applied to the data: constant and variable.

2.2.4 Constant filters

The constant filters are not just a part of the process, and they are a crucial element in ensuring the

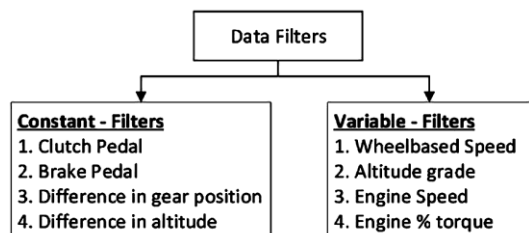


Fig. 4 — Classification of data filters.

vehicle's mobility without any change in driving parameters. For instance, when the vehicle is not stationary and moving at a constant speed without any change in gear position, the constant filters come into play. The data range for these filters is given below in Table 4 with reasons. Figure 5 (a-d) shows the variations of the data points collected for each feature plotted over time. These variations can affect the mass

estimation; hence, the constant filter is of utmost importance in removing unnecessary data points from the data points.

2.2.5 Variable filters

The constant filter does a great job of eliminating the idle data points of the operating parameters, but it's not enough. The data still shows noise due to the vehicle's different operating conditions, like the

Table 4 — List of constant filter variables and their value range used for mass estimation.

Constant filter	Value	Impact
Clutch Pedal, C(t)	= 0	The sensor value will return 0 – when the clutch is not applied and 1 – when it is used. Provide the data points without clutch change to ensure engine and gear ratios.
Brake pedal, B(t)	=0	The sensor value will return 0 – when the clutch is not applied and 1 – when it is used. The data points with values equal to 0 are filtered to ensure the vehicle is not stationary.
Difference in Gear position, G(t) – G(t-1)	=0	Each data point will provide the gear position value. The difference between the previous and current data point's gear position values is confirmed as 0 to ensure the vehicle moves at a constant speed.
Different in Altitude, A(t) – A(t-1)	0 > A(t) – A(t-1) < 6	The GPS sensors will provide the altitude data of the moving vehicle to ensure it is not moving in an inclination or deceleration type of road. The difference in altitude between 0 and 6 provides the vehicle's movement at a constant altitude.

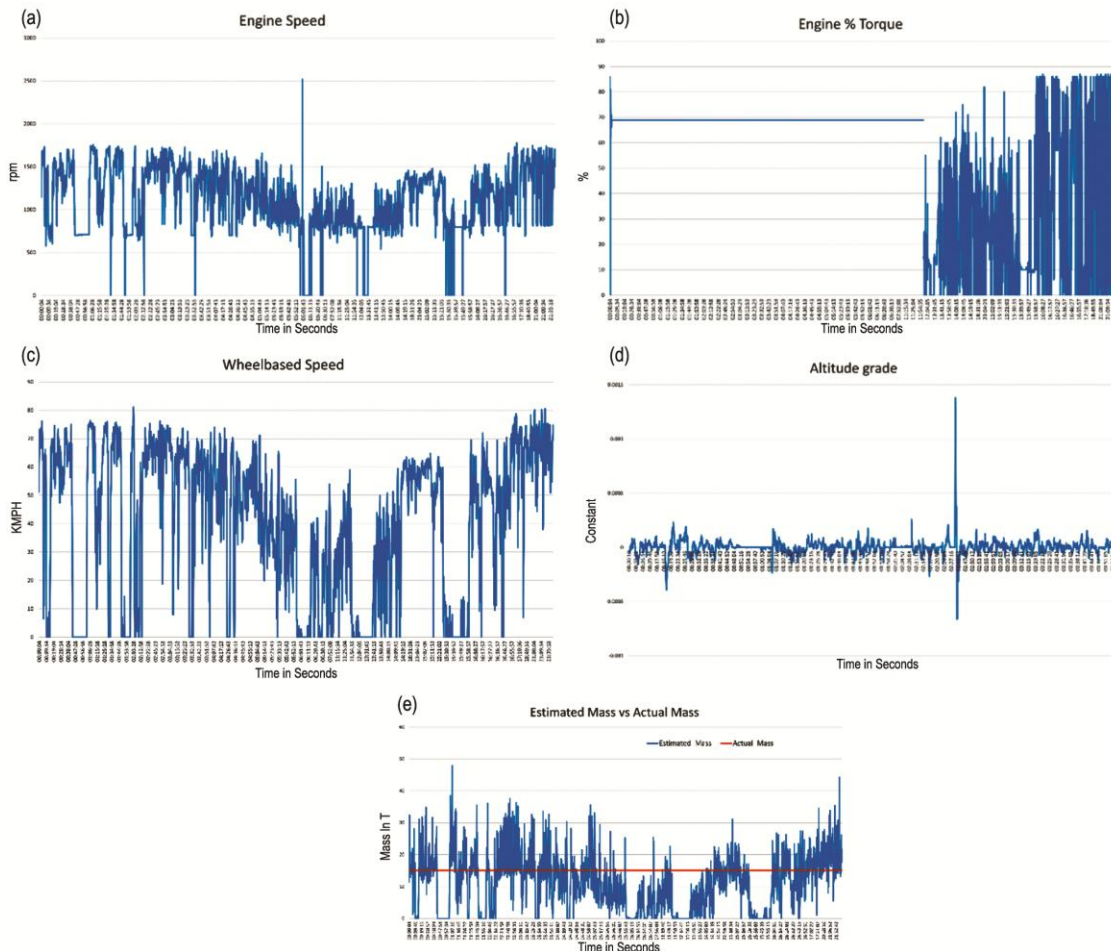


Fig. 5 —The graphs show the filtered vehicle influencing parameters spread over time: (a) engine speed, (b) engine percentage torque, (c) wheel-based speed, (d) altitude grade, and (e) comparison of estimated mass vs. actual mass.

vehicle running on a bumpy road, different altitudes, etc. This is where the variable filter comes in. The variable filter constraint shown in Table 5 is applied to the operating data points to eliminate the outlier data and focus only on the relevant data points required for the mass estimation. The variable filter values vary based on vehicle segment and engine power. Thus, this constraint is called a variable filter. After applying both filters, a minimum of 5 and 50 operating parameters of data points are required for mass estimation. If less than 5 data points are available, the trip will be discarded from mass estimation.

**2.3 Mass validation**

After filtering by the variable filter, the operating parameters are ready to be deployed in the vehicle dynamics system model formula Equation (14). The estimated mass for each data point is calculated and stored against each operating parameter. Thus, each trip can have 5 to 50 data points with estimated mass. To find the average operating mass of the vehicle for the entire trip, a statistical method is used to find the best operating mass of the vehicle by using the Z-score technique.

Thus, for each trip data, the z-score is computed for a set of trip data for the trip ID using Equation (15) for each data point calculated, and the estimated mass with a better z-score is taken as the trip’s final estimated mass. Table 6 shows the sample data of

how the z-score for data points for a trip is computed and the final estimated mass is selected.

$$Z = \frac{x - \mu}{\sigma} \dots (15)$$

Where x is the observed value, μ is the mean, and σ is the standard deviation of the sample.

The system model was validated using trip data with different driving and load conditions. A 19-tonne 4x2 haulage truck model with a payload capacity (loading capacity) of 11 tonnes was used for this validation. For this validation, four vehicles of the same model were used, termed VDC012, VDC015, VDC016, and VDC017. These vehicles were loaded at less than 100% capacity and more than 100% capacity and ran on actual roads for a minimum of 60 to a maximum of 180 minutes trip. 40% rated load trips and 60% overload trips were conducted, and the data were collected. The vehicle was driven by drivers who used to drive regular vehicles to ensure the driving behaviour was not modified.

The collected data were processed using the system model to get the estimated mass for the trip data. They later computed the z-score to identify the final estimated trip data, with a cumulative accuracy of 87%. The sample of the processed data with accuracy computation and type of loading condition were given in Table 7. Figures 6 and 7 are plotted to show the actual mass vs. estimated mass for different loading conditions given in Tonne and the system model

Table 5 — List of Influencing variables and their value range

Influencing Variable filter	Value	Impact
Wheelbased speed, W(t)	40 – 75	Ensure the vehicle is in the constant speed range.
Altitude grade, D(t)	0 – 0.015	The vehicle is not driven on a slope.
Engine speed, E(t)	>1200	The engine is not idling and running at a constant speed
Engine % torque, M(t)	40 – 90	To ensure the vehicle is running at constant power

Table 6 — Sample data shows the Z-score computed for a trip id

Trip ID	Vehicle Type	Vehicle Model	Torque	Engine Speed	Wheelbased Speed	Frictional Torque	Altitude	Estimated GVW	standard deviation, σ	z-score
T25252	Haulage	1920	81	1146	73.4	9	95.5	11.4	0.25	-0.671
T25252	Haulage	1920	71	1620	68.77	8	103.26	11.3	0.36	-0.805
T25252	Haulage	1920	70	1684	70.34	9	104.26	12.9	1	1.3423
T25252	Haulage	1920	66	1633	67.71	9	99.37	11.4	0.25	-0.671
T25252	Haulage	1920	85	1572	71.01	9	93.279	12.5	0.36	0.8054
							mean (μ)	11.9	0.7441	

Table 7 — Sample data shows the various load conditions and accuracy calculated using the system model.

Vehicle ID	Trips	Load Condition	Pay Load %	Actual Mass (in T)	Estimated Mass (in T)	Accuracy %
VDC017	T2164	Overload	123%	13.5	10.935	81%
VDC015	T2896	Rated Load	65%	7.2	7.416	103%
VDC012	T3244	Overload	109%	12	10.68	89%
VDC016	T41232	Rated Load	73%	8	5.2	65%
VDC015	T5477	Overload	129%	14.2	15.194	107%
VDC012	T67867	Overload	106%	11.7	10.179	87%
VDC015	T7253	Overload	118%	13	10.166	78%
VDC016	T825	Rated Load	91%	10	7.72	77%
VDC016	T9474	Rated Load	45%	5	5.4	108%
VDC016	T1034	Overload	100%	11	10.032	91%

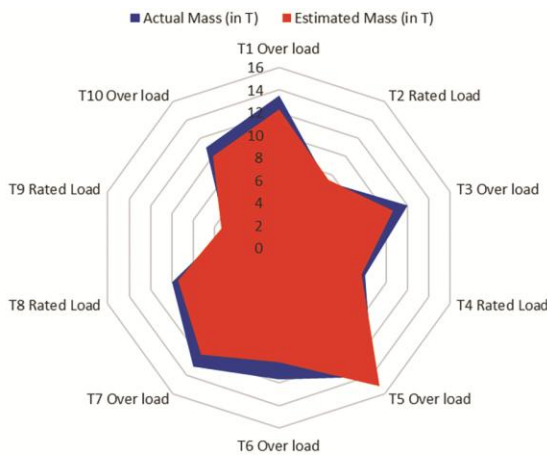


Fig. 6 — The graph compares actual vs. estimated mass with different loading patterns.

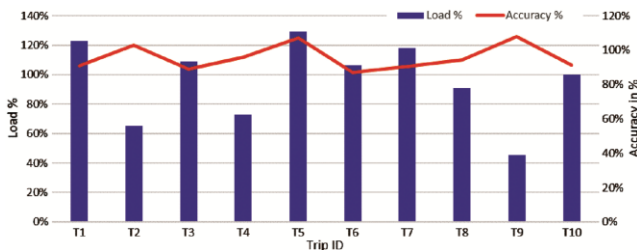


Fig. 7 — The graph shows the Payload % vs. System model accuracy %.

accuracy % plotted against the payload %, respectively. The estimated mass of the vehicle for various trips is stored in the table, which is used as master data for training and testing the MLP model, which is used to predict the vehicle’s mass.

**2.4 Mass prediction using ML model**

The mass estimated through the dynamics system model provided a better result. Still, a machine learning model is needed to improve the agility of the process. As the dynamics system model requires more time to process the trip data and estimate the mass, the introduction of a deep learning machine learning

model will help to predict the mass faster, as there are more than 1 Lakh trip data from more than 80,000 vehicles per day needs to be predicted and stored, the need for machine learning model arise, and this MLP model helps to predict the mass of the vehicle faster. Moreover, this two-step approach provides a cushion for the team to continuously improve the dynamics model accuracy based on various practical factors and thus retrain the model to predict more accurately.

**2.4.1 Algorithm selection**

Selecting the MLP algorithm for a regression problem offers several advantages supported by research findings and real-world applications. Below are some key reasons, along with relevant citations:

**Nonlinear regression capability:** MLPs are well-suited for capturing complex nonlinear relationships in data, making them suitable for regression tasks where the underlying relationships are not linear. According to a study by Xu et al.<sup>39</sup>, MLPs have demonstrated superior performance over linear regression in capturing and modeling nonlinear patterns in various real-world datasets.

**Flexibility and adaptability:** MLPs offer great flexibility in their architecture. Researchers have shown that adjusting the number of hidden layers and neurons and selecting appropriate activation functions enables MLPs to adapt to the complexity of the regression problem. Yang et al.<sup>40</sup> emphasize this adaptability in their research, discussing how fine-tuning MLP architectures can improve regression accuracy.

**Automatic feature learning:** MLPs can automatically learn relevant features from the input data without extensive manual feature engineering. This feature learning capability reduces the effort required in data preprocessing and allows MLPs to handle high-dimensional datasets effectively. Zhang et al.<sup>41</sup> demonstrated the efficacy of MLPs in

automatically learning features in their study on regression problems in medical image analysis.

**Generalization and robustness:** MLPs have demonstrated robust generalization capabilities, which means they can perform well on unseen data. Research by Li et al.<sup>42</sup> highlighted the robustness of MLPs in regression tasks involving noisy datasets, showing their ability to handle noise and produce reliable predictions.

**Ensembling for improved performance:** MLPs can be used in ensemble methods, where multiple MLP models are combined to enhance predictive accuracy and reduce overfitting. According to a research paper by Zhang and Wang<sup>43</sup>, ensemble learning with MLPs has significantly improved regression performance, mainly when dealing with complex datasets.

Thus, the Multilayer Perceptron algorithm is a powerful and versatile choice for regression problems, supported by its ability to capture nonlinear patterns, automatic feature learning, and flexibility in

architecture. Its widespread use in various domains, coupled with the advantages highlighted in the research literature, underscores its relevance and importance in regression modeling. Table 8 shows the pros and cons of comparing the MLP algorithm with other ANN algorithms.

**2.4.2 Feature and target definition**

The five mass-influencing parameters used for the estimation are the input to the MLP algorithm. They are engine speed  $E(t)$ , wheel based speed  $W(t)$ , engine torque  $Q(t)$ , vehicle model ( $M$ ), and altitude grade  $D(t)$ . The vehicle ( $m$ )'s mass is defined as the model's target. As shown in Figure 8, the MLP model is represented by one input layer with four inputs, one hidden layer with 100 nodes, and one output layer.

The code was executed in Spyder web integrated development environment (IDE). The data used for training is the model of the estimated mass using the dynamic system model for unique VIN and trip ID for each segment and model of the truck. In this case, the

Table 8 — Comparison of various ANN algorithms

Algorithm	Pros	Cons	Use Cases
Multilayer Perceptron (MLP)	Nonlinear regression capability	Sensitive to hyperparameters	General regression tasks, complex nonlinear patterns
	Flexibility in architecture	Prone to overfitting	Feature learning, high-dimensional datasets
	Automatic feature learning	Interpretability (for deeper models)	Ensemble learning for improved performance
	Robust generalization Availability of libraries and frameworks	Computationally intensive	
Convolutional Neural Network (CNN)	Excellent for image data	It may require large datasets	Image regression tasks, spatial dependencies
	Feature extraction capabilities	Complex architecture	Computer vision, pattern recognition
	Parameter sharing for reduced complexity	Training time	
	Hierarchical feature representation Robust to spatial transformations		
Long Short-Term Memory (LSTM)	Sequential data processing	Sensitive to hyperparameters	Time series regression, natural language processing
	Memory for temporal dependencies	Gradient vanishing/exploding	Temporal patterns, sequential data
Recurrent Neural Network (RNN)	Variable-length input sequences	Computationally intensive	Sentiment analysis, speech recognition
	Handling time series data	Longer training times	
	Natural language modeling		
Radial Basis Function Network (RBFN)	Fast training and convergence	Sensitivity to RBF centers	Approximation of smooth functions, radial symmetry
	Local learning of radial basis functions	Limited scalability for high-dimensional data	Function approximation, interpolation
	Interpolation and extrapolation capabilities	Complexity increases with the number of RBFs	
	Suitable for sparse data Robustness to outliers		

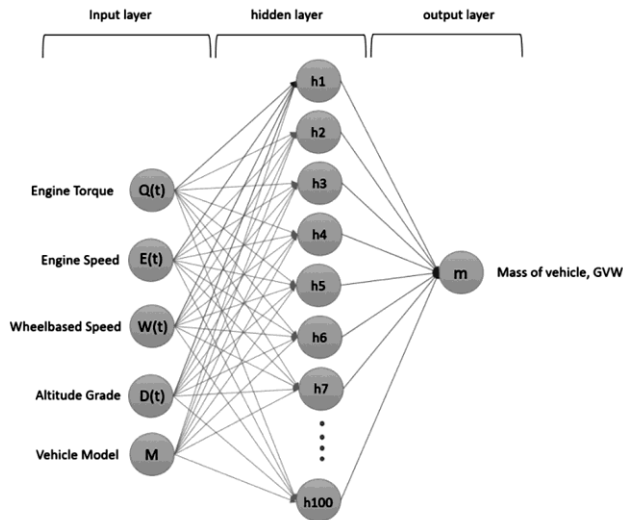


Fig. 8 — Structure of Artificial Neural Network MLP network.

data from a medium-duty truck of 19-tonnage GVW vehicles is used to train the model. There are 10,000 trip data available for the vehicle, which are run in different load and road conditions, in which 8,000 trip data were used for training, and 2,000 trip data were used for testing the model. The MLP model uses Tensor Flow and Scikit libraries to train the model. The data used to train the model will be scaled using function scale datasets. This is highly required to improve the fastness of the model as the scaling will improve the convergence to the global optimal value for the loss function optimization function. Scikit Learn's Standard Scaler class is used in this model, which performs z-score normalization on the predicted mass data for multiple values. This method subtracts each data from its mean and divides it by the standard deviation. Thus, the data for training will be trained using a scaled dataset. The ANN MLP model uses `asci-kitlearns` neural network MLP Classifier function with one hidden layer. The hidden layer consists of 1024 hidden units/ neurons with the Rectified Linear Units (ReLU) activation function. The output layer consists of one unit with a linear activation function.

#### 2.4.3 Trained ANN model

The built model must be trained with the estimated mass dataset, which is compiled using training configuration. The total dataset is split into train data, 80%, and test data, which is 20%. The mean squared logarithmic loss (RMSLE) is used as a loss function to calculate the loss of the predicted output.

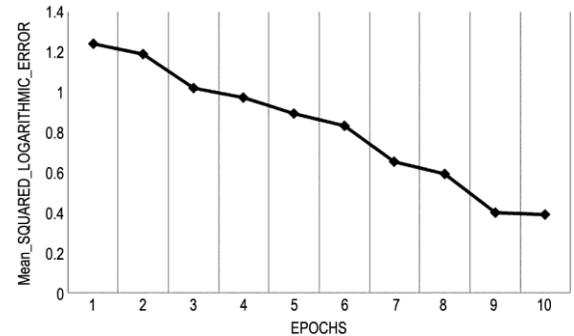


Fig. 9 — Logarithmic error decreases as the epochs increase.

The hyperparameters need to be configured to train the model for better accuracy. Thus, ten epochs to run are chosen. The training dataset contains 8,000 records to train with ten epochs, and the data is split into the batch size of 2000 records. Thus, four iterations are required to complete one epoch. At the end of the training, the learning convergence is plotted for every epoch using the plot function in the `sklearn` toolkit library. As per Figure 9, the logarithmic error value decreases as the epochs increase, ensuring the model is a good fit and that no model overfits in training. Thus, the model's training is completed, and the testing of the model will be executed with the identified 2000 records for test validation. The model will then be trained and tested. Thus, the firm ANN MLP model is ready for deployment to field testing with actual data.

### 3 Results and Discussion

The firmed MLP model is validated using three datasets to measure the model's accuracy in various scenarios since this research aims to find the mass estimation of the vehicle and classify each trip based on the operating load. These three scenarios become more appropriate: a) No load, b) Partial load, and c) Overload condition. The vehicle that ran with all these three loading conditions for 30 minutes on a normal road with practical driving conditions was measured, and the data was passed into the MLP model to predict the mass. For validation purposes, 50 data points that adhere to the constant and variable filters were taken in each condition and validated. The results and inference are discussed below. The summary of different loading conditions with MSE and % of the time the error is below the 10% range is summarized in Table 9.

#### 3.1 No load condition

In the no-load condition, the vehicle is not loaded with any mass but has its kerb weight, also called

Table 9 — Performance metrics of a mass prediction model with various loading conditions

Condition	GVW (in T)	ULW (in T)	Payload (in T)	Actual Mass (in T)	RMSE	% of Time (Error +/-<10%)
No Load	16	7.5	0	7.5	0.985	78%
Partial Load	16	7.5	5.2	12.7	0.648	86%
Overload	16	7.5	13.5	21	0.003	92%

unladen weight (ULW). The prediction model for the no-load condition was not performing to the required levels, the RMSE was 0.985, and the % of the time the error was less than 10% is 78% in the 50 data points, as shown in Fig. 10 (a). This error occurred because the vehicle influencing operating parameters underperformed the optimized ranges. Most of the time, the engine torque and speed could be utilized more, which is the primary reason for the predicted error to underperform. Concerning the objective of this research, this will not impact the purpose, but there needs to be an improvement to relook the optimization process to consider the no-load conditions during mass estimation.

**3.2 Partial load condition**

During partial load conditions, the vehicle was loaded with 5.2 tons of payload to ensure its mass was 12.7 tons. The prediction model performed to predict the RMSE as 0.648, which is an improvement compared to no load condition, and the % of the time the error is less/more significant than 10% is 86% w.r.t. 50 data points as shown in Fig. 10 (b). The results were encouraging as the vehicle was driven in the same road conditions.

**3.3 Overload condition**

In this condition, the vehicle is loaded with 13.5 tons of payload to make the actual mass 21 tons, an overload condition higher than the defined payload carrying capacity and GVW. The MLP model performed excellently to predict the mass with an RMSE of 0.003, and the % of time error is lesser/greater than 10% as 92% w.r.t. 50 data points as shown in Fig. 10 (c) improved the confidence of the MLP model. These results encourage the purpose of this research and ensure the use of the model further in real-time vehicle models and start recording the vehicle’s mass.

These results show that the MLP model performs exceptionally well when the load conditions are high since the operating parameters are constant and don’t deviate more due to other noise. The operational parameters vary in partial and no-load conditions, which disturbs the model’s accuracy. This learning

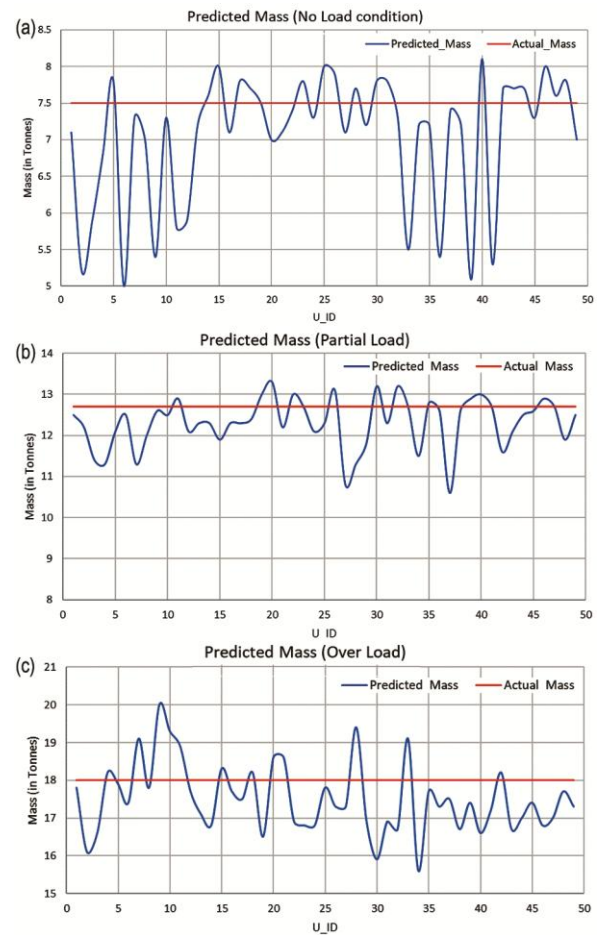


Fig. 10 — Graphs showing the mass prediction for various loading conditions using the MLP model (a) Predicted Mass using MLP model with no load condition, (b) Predicted Mass using MLP model with partial load condition, and (c) Predicted Mass using MLP model using overload condition.

helps ensure these observations are incorporated into the future model and make it more robust. The data from the model is beneficial for design optimization and understanding the various loading conditions for different vehicle applications.

**3.4 Future works**

The development of this model opened many applications in the commercial vehicle industry, a few of which are listed below,  
 a This model can be used to develop new product service systems for truck owners.

- b The model on further fine tuning can be used to calculate the dynamic load of the truck to know the loading and unloading patterns for various applications like delivery trucks, tankers, etc.,
- c For Passenger vehicles like buses, there is a vast potential to know the loading pattern during peak and non-peak hours for fuel efficiency and operating load optimization.

### 3.5 Improvements in the model

In this methodology, many influencing parameters affect the model's performance, which needs to be improved in future works.

- a Although the dynamics system model is a generic method to estimate mass, the algorithm's constants depend on the type of vehicle, running conditions, etc. Thus, the system model should be programmed dynamically to consider the constants based on the vehicle model and running condition.
- b The application of data filters to get the vehicle operating parameters based on practical driving conditions must be further optimized to cover all driving conditions.
- c Due to operational difficulties, the MLP model has not been tested with various road conditions, but it can be tested in future work to get better training for the model.

The MLP model is based on the artificial neural network family; this method was chosen because it has a better learning rate when back propagation techniques are used. Thus, the algorithm's accuracy and agility are better. Still, there is scope to evaluate other regression techniques for this data and compare the results for improvements.

## 4 Conclusion

The automotive sector was an early adopter of several new technologies, such as big data, IoT, embedded software, artificial intelligence and machine learning (AI/ML), deep learning, automation, etc. Compared to passenger vehicles, the commercial vehicle space within the automobile industry has more distinct challenges. Even though the research began with the goal of determining the mass of the trip operated by the vehicle, this model has provided confidence and potential to explore the possibility of identifying the real-time mass of the vehicle within the trip based on loading and unloading in the truck. To determine the mass, this two-step methodology combines the mathematical model and the deep learning machine learning model; this model gives leverage for future research efforts to improve

the model in step 1 and train the model with different deep learning algorithms in step 2. The final model even predicts the mass for each trip with an accuracy of 90% in the MLP model and 87% in the dynamic system model; however, practical constraint scenarios must be added to the model in the future to make the model comprehensive. The benefit of this model is that it predicted mass with no additional sensor expense because all the data used in this model were existing EDC data stored in the cloud. The proposed model was tested on a medium-duty truck, but it should be evaluated on other segment trucks, such as tippers and tractors, which carry heavier loads. This model will be an integral part of the digital twin framework to predict the real-time mass of the vehicle based on which many parameters can be controlled or optimized, or real-time warnings can be provided. The real-time mass can be used to design various vehicle systems and configurations.

## References

- 1 Lin N, Zong C & Shi S, *Energies*, 11 (2018) 3042.
- 2 Hayashi K, Shimizu Y, Dote Y, Takayama A, & Hirako A, *IEEE Trans Control Syst Technol*, 3 (1995) 49.
- 3 J Ma, J Hu, E Leslie, F Zhou, P Huang & J Bared, *Transp Res, Part C, Emerg Technol*, 100 (2019) 125.
- 4 He C, Maurer H & Orosz G, *J Comput Nonlin Dyn*, 11(2016) 061011.
- 5 Y Liu & W Wang, *Sensors* 21(18)(2021) 1.
- 6 Y Zhang & J Yin, *Research on ABS logic sliding mode control method based on road surface recognition*, paper presented at IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, Chengdu, 2019.
- 7 Ji FZ, Pan Y, & Zhou Y, *Vehicle Syst Dyn*, 58 (2019) 144.
- 8 Huang Y, Wang H, & Khajepour A, *J Power Sources*, 341 (2017) 91.
- 9 Daily R & Bevely D, *IEEE Trans Ind Electron* 51 (2004) 270.
- 10 Bae HS, Ryu J, & Gerdes JC, *Road grade and vehicle parameter estimation for longitudinal control using GPS*, paper presented at IEEE conference on intelligent transportation systems, Oakland, CA, 2001.
- 11 Han KJ, Kim IK, & Jo HY, *Proc IMechE, Part D J Automobile Eng*, 223 (2009) 167.
- 12 ZHENG YiJia, SUN Jian & NIAN Guangyue, *China Journal of Highway and Transport*, 35(5) (2022) 243.
- 13 Erik Wilhelm, Lennon Rodgers & Raffaele Bornatico, *Real-time electric vehicle mass identification*, paper presented at EVS27 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, Barcelona, Spain, 2013.
- 14 Ahmed A, Santhosh J & Aldbea F, *Vehicle dynamics modeling and simulation with control using single track model*, paper presented at IEEE International Women in Engineering (WIE) Conference on Electrical and Computer Engineering (WIECON-ECE) Bhubaneswar, 2020.
- 15 Yazdani BB & Frey HC, *Atmos Environ*, 85 (2014) 179.

- 16 Bian J & Qiu B, *Proc I Mech E, Part D J Automobile Eng*, 232 (2018) 1736.
- 17 Zhang X, Xu L, & Li J, *Real-time estimation of vehicle mass and road grade based on multi-sensor data fusion*, paper presented at IEEE Vehicle Power and Propulsion Conference (VPPC), Beijing, China, 2013.
- 18 Rajamani R & Hedrick JK. *IEEE T Contr Syst T*, 3 (1995) 86.
- 19 Zhang Y, Zhang Y, Ai Z, Feng Y, Zhang J & Murphey Y, *IEEE Trans Veh Technol*, 68(11) (2019)10627.
- 20 Cai L, Wang H, Jia T, Peng P, Pi D & Wang E, *Proc IMechE, Part D J Automobile Eng*, 234(2–3) (2020)378.
- 21 Seungki K, Kyungsik S, Changhee Y & Huh K, *Int J Automot Technol*, 18(6) (2017)1077.
- 22 Zhao J, Li Z, Zhu B, Li Y & Sun Y, *Zhongguo Gonglu Xuebao*, 32(12) (2019)58.
- 23 Sun Y, Li L, Yan B, Yang C & Tang G, *Mech Syst Signal Process*, 68–69(2016) 416.
- 24 Hu M, Luo Y, Chen L & Li K, *Jilin Daxue Xuebao (Gongxueban)*, 48(4)(2018) 977.
- 25 Chu W, Cao K, Li S, Luo Y & Li K, *Vehicle mass estimation based on high-frequency-information extraction*, paper presented at 7th International Federation of Automatic Control (IFAC) Symposium on Advances in Automotive Control, Tokyo, 2013.
- 26 Shih C, Huang P, Yen E & Tsung P, *Vehicle speed prediction with RNN and attention model under multiple scenarios*, paper presented at IEEE Intelligent Transportation Systems Conference (ITSC)IEEE, New Zealand, 2019.
- 27 Ziółkowski J, Oszczypała M, Małachowski J, & Szkutnik Rogoż J, *Energies*, 14(9) (2021) 2639.
- 28 Torabi S, Wahde M and Hartono P, *Road grade and vehicle mass estimation for heavy-duty vehicles using feedforward neural networks*, paper presented at 4<sup>th</sup> International Conference on Intelligent Transportation Engineering (ICITE), IEEE, Singapore, 2019.
- 29 Korayem A, Khajepour A & Fidan B, *IEEE Trans Veh Technol* 69(11)(2020) 12536.
- 30 Kenneth M J, Ilmar F S, Line KH C, Søren Theodorsen & Harry J P C, *Mechanical Systems and Signal Processing*, 162 (2022) 107982.
- 31 Yu Z, Hou X, & Leng B, *Auton Intell Syst*, 2 (4) (2022) 1.
- 32 Eagon M, Fakhimi S, Pernsteiner A & Northrop WF, *Mass Detection for Heavy-Duty Vehicles using Gaussian Belief Propagation*, paper presented at IEEE Intelligent Vehicles Symposium (IV), Aachen, Germany, 2022.
- 33 Mutta S, Subramanian S C, Gunti S, Sathiyarayanan M, Shankar V & Parag D, *Model-Based Road Slope and Mass Estimation of Road Vehicles*, paper presented at IEEE Transportation Electrification Conference (ITEC-India), New Delhi, India, 2021.
- 34 Wang B, Wang H, Wu L, Cai L, Pi D & Wang E, *Proc IMechE, Part D J Automobile Eng*, 234(10-11) (2020) 2429.
- 35 Cai L, Wang H, Jia T, Peng P, Pi D & Wang E, *Proc I Mech E, Part D J Automobile Eng*, 234(2-3) (2020) 378.
- 36 Lin N, Zong C & Shi S, *Energies* 12 (2019) 52.
- 37 Sahlholm P & Johansson K, *Control Eng Pract*, 18(11), (2010)1328.
- 38 Boucher C & Noyer J, *Sensors* 18(1)(2018) 25.
- 39 Xu C, Wang Y, Xu C & Zhang X, *IEEE Access*, 8 (2020) 22294.
- 40 Yang W, Liu X & Zhang Y, *Applied Sciences*, 11(8) 20213733.
- 41 Zhang L, Tan W, & Zhang L, *Complexity*, 2020, 1.
- 42 Li Q, Zhao K, Zheng Y & Zhang Z, *Journal of Ambient Intelligence and Humanized Computing*, 13(3) (2022) 3039.
- 43 Zhang J & Wang L, *Neural Computing and Applications*, 33(3) (2021) 951.