CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Data-Driven Digital Twin for EV Energy Consumption Prediction

Master's thesis in Computer science and engineering

Huitong Gao
Tianshuo Xiao

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2024

Master's thesis 2024

Data-Driven Digital Twin for EV Energy Consumption Prediction

Huitong Gao
Tianshuo Xiao

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2024

Data-Driven Digital Twin for EV Energy Consumption Prediction
Huitong Gao
Tianshuo Xiao

Supervisor: Devdatt Dubhashi, Data Science and AI
Advisor: Utsav Khan, Zeekr Technology Europe
Examiner: Devdatt Dubhashi, Data Science and AI

Master's Thesis 2024
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Displayed in the center console of the Zeekr 001

Typeset in LaTeX
Gothenburg, Sweden 2024

Data-Driven Digital Twin for EV Energy Consumption Prediction
Huitong Gao
Tianshuo Xiao
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Driven by environmental requirements and new technologies, electric vehicles (EVs) are rapidly gaining momentum due to their environmental and economic advantages. EVs are gradually replacing fuel-powered cars to become main mode of transportation. However, a major obstacle to the widespread adoption of EVs is range anxiety. Drivers worry that their vehicles do not have enough range to reach their destinations.

To alleviate range anxiety, it is essential to accurately predict the energy consumption of EVs between the departure and the destination. This Master Thesis proposes data-driven methods for predicting EVs energy consumption. The data used included vehicle data provided by Zeekr Technology Europe AB and environmental data provided by external APIs. Vehicle and environmental data were processed into distance sequence data. When selecting energy prediction parameters, it was found that speed is required as a feature for energy prediction. Therefore, energy prediction was divided into two parts: speed prediction and energy prediction.

Long short term memory (LSTM) model was used for speed prediction. For predicting energy consumption, a vehicle dynamics model was built as baseline. Basic machine learning models (linear regression, decision tree, random forest, and k-nearest neighbor.) and convolutional neural network with long short term memory (CNN-LSTM) model were used to predict energy consumption.

The results of speed prediction were better than those of the map API and the existing model. The conclusion of the energy prediction was that all machine learning models performed better than the vehicle dynamics model. For short distances, the decision tree model provided the best predictions, while for long distances, the CNN-LSTM model offered the best predictions.

Keywords: energy prediction, speed prediction, machine learning, electric vehicle (EV), convolutional neural network with long short term memory (CNN-LSTM).

Acknowledgements

We would like to express our special thanks to our supervisor at Zeekr Technology Europe, Utsav Khan, for providing professional guidance and practical experience throughout our thesis. We are also deeply grateful to our supervisor and examiner at Chalmers, Devdatt Dubhashi, for his exceptional guidance and assistance with our thesis. Special thanks go to Hamza Bouchouireb, Marcus Andersson who offered technical support and assisted with data collection and analysis. We would like to thank our manager Fangkun Lindström for her care and concern. We would like to extend thanks to Jaewoo Joung and all the employees at Zeekr for their support. Their cooperation and support were essential for us to complete this thesis successfully.

Huitong Gao, Tianshuo Xiao Gothenburg, 2024-06-16

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

Electric vehicles (EVs) have been developing rapidly in recent years and gradually replacing internal combustion engine (ICE) vehicles as the dominant mode of transport. However, estimating the remaining range has always been a pressing issue due to the different weather or traffic conditions faced during driving. To alleviate range anxiety [1] and increase confidence in current predictions of the remaining driving range on board, a data-driven approach is necessary to estimate the energy consumption and range of electric vehicles. This approach will also improve trip planning capabilities.

Range anxiety poses a challenge to the widespread adoption of EVs, stemming from drivers' uncertainty about their vehicle's state of charge (SOC) and the energy needed to reach their destination. Many existing estimation methods for these parameters rely on simplified models with numerous assumptions, leading to potentially significant errors, especially when dynamic and environmental conditions [2] are not taken into account.

A significant barrier to predicting energy consumption lies in the analytical modelling of a variety of factors, including road characteristics (e.g. traffic signals and road type), weather conditions and time of day, all of which affect traffic flow [3]. Data-driven modelling can therefore provide more accurate solutions. In addition, individual driver behaviour plays a crucial role in energy consumption. Capturing these patterns from driving data can greatly improve the accuracy of energy consumption predictions.

1.2 Literature review

Surveying the relevant literature, the prediction of EV energy consumption can be categorised into two approaches, which are model-based and data-driven vehicle methodologies. Some articles have also proposed the use of hybrid methods to Physics Augmented Models with data-driven approaches.

The fundamental approach to predicting energy consumption using the vehicle model-driven method involves developing a model that simulates the dynamics of an electric

vehicle (EV) during driving. This simulation is crucial for assessments like those conducted under the Worldwide Harmonized Light Vehicles Test Procedure (WLTP), which is a standardized test protocol primarily used in Europe. WLTP measures fuel consumption, CO2 emissions, and pollutant emissions of light vehicles, including EVs under controlled conditions. Specifically for electric vehicles, it evaluates the state of battery charge and consumption, as well as the battery's range under various driving scenarios, providing a comprehensive picture of vehicle performance [4].

In the article by Wu et al. [5], an analysis is carried out on the relationship between EV power, vehicle speed, acceleration, and road gradient. By modelling the resistance encountered by EVs during operation, the study proposes an analytical model of EV energy consumption estimation.

The article by Miri et al. [6] focusses on the modelling and estimation of EV energy consumption. They proposed that the choice of method for estimating energy consumption depends on the target application. In general, statistical and computational models require more computational work than analytical models. However, they are more accurate because they are based on data analysis and probabilistic predictions. Additionally, analytical models can only reflect changes in vehicle behaviour because they are based on vehicle dynamics and physical models.

However, existing vehicle model-driven methods focus predominantly on the physical dynamics and internal factors of EVs (vehicle design parameters, efficiency and inertia of vehicle components, etc.), which overlook crucial contextual variables such as weather conditions, temporal variations in traffic flow, and road conditions. This limitation undermines the predictive accuracy of the models, particularly when faced with various environmental and traffic scenarios. Consequently, if these external factors are not taken into account, it will lead to unsatisfactory model predictions, affecting the efficacy of EV performance estimation in real-world settings.

In the Cauwer et al. [7] article, they use a data-driven approach for the energy consumption prediction. The article's prediction model was divided into two parts, the first part was the prediction of the car's dynamics values, using a neural network to predict the kinetic and potential energy. The predicted values were then used to predict the energy consumption. In the input parameters, the article considered the effects of external factors, such as weather change and traffic flow, which made the predictions of the model more accurate.

The article by Qi et al. [8] proposed to obtain an accurate link-level (line-level refers to situations that occur on segregated sections of the transport network) energy estimation model for electric vehicles by link-level energy estimation, which decomposed the energy consumption under real-world traffic congestion through positive kinetic energy (PKE) and negative kinetic energy (NKE).

The articles and reports discussed above focus on predicting energy consumption in electric vehicles (EVs). Energy consumption analysis and estimation of energy consumption for EVs is critical for various applications centered on EVs, like route planning, charging station planning. They involve developing models based on vehicle

dynamics or physics principles and validating predictions against testing standards. Analytical models, such as those in the studies by Wu and Miri, offer accuracy but demand significant computational resources. However, they often overlook external factors such as weather and traffic conditions, which affect the predictive accuracy. Data-driven approaches, such as those in Cauwer's work, incorporate these factors, improving model precision. Qu's article proposes link-level energy estimation, and decomposing consumption under real-world traffic conditions. Despite advances, existing methods struggle to fully account for contextual variables, limiting their applicability in the real world and their predictive efficacy.

However, the limitation of the current approach is that in the data-driven based approach, the data collected is limited to a specific road or a zone, which can only achieve short-term energy consumption prediction, and cannot be used for long-term prediction to cover the whole country or different driving scenarios. In addition, while most models predict energy consumption based on changes over time, in practical applications, it is important to consider the impact of the distance between two locations on energy usage.

In this project, we compare different features and filter them to include internal inputs and external inputs, and compare different machine learning methods to make predictions of energy consumption. We take driving resistance into account as an important parameter when making energy predictions. We propose two methods for energy prediction, one is to predict instantaneous energy consumption, the other method is to directly predict the total energy consumption. In the total energy prediction, we used a deep learning approach for prediction (long and short term memory network), which is able to process the sequence problem better. This approach proves to be superior to traditional data-driven methods, improving accuracy and adaptability by comparing error matrix.

1.3 Aims and objectives

Current industry standards for such predictions typically rely on physical resistance models. These models are effective in many situations, but still have some obvious limitations, including often insufficient consideration of variable environmental factors and a tendency to oversimplify driver behavioural dynamics. To address these limitations, this thesis proposes a hybrid modelling approach that not only combines the predictions of physical resistance models, but also integrates a wide range of environmental and travelling data.

The overall goal of this thesis is to work with Zeekr Technology Europe to investigate the use of a data-driven method to develop two models, one for speed prediction and the other for prediction of EV energy consumption.

The speed prediction model focuses on the physical characteristics of vehicle operation, primarily using external data inputs to predict vehicle speed. The energy consumption prediction model, on the other hand, shifts the focus to relevant human factors, especially predicting the energy consumption of EVs based on user driving behaviour. As speed is an important parameter of energy consumption, we

need to predict the speed first and take the predicted speed as a feature input to the energy prediction model. By splitting the models in this way, each model can be specialised and optimised for its own prediction task. When combined these two models, they can provide predicted values of vehicle energy consumption under real-world conditions.

All onboard data was collected from Zeekr's test vehicles, and we built datasets and trained machine learning models based on these data. These models can provide improved energy estimates compared to the model that was currently in use.

1. **Vehicle data acquisition** from Influxdb. All vehicle data in the thesis comes from Zeekr's experimental vehicles and the driving range was concentrated in the Nordics. The data contains information about the vehicle's location, speed, and battery power information while driving. These data are used to calculate the dynamic information of the vehicles.
2. **Environmental factor data acquisition** from external. External environmental factors are important factors affecting energy consumption and we obtain this data from different APIs. The data of environment contains weather, wind speed, wind direction, precipitation etc. which will be an important part of the dataset.
3. **Process vehicle data and API data** according to the sampling rate of battery information. Because of the different sampling frequencies of the sensors on the vehicle, we need to pre-process these data to get the final synchronized data. After synchronising the data, we can downsample the data and get the external environment data through API.
4. **Build a vehicle dynamics model** based on vehicle parameters and data acquired on-board and environmental data.
5. **Train the machine learning models** based on the vehicle dynamics data, on-board data and external environmental data. We need to build two models, one to predict the speed, which specific focuses on understanding and forecasting driver behavior within the given road context. In addition, one of the goals of the speed model is to replace the reliance on the API for speed estimation and to create our own internal model. The other one is used to predict the energy consumption of the EVs, which focuses on learning and predicts the physical behavior of the car, taking into account the driving strategy and various environmental factors. The training data for the speed prediction model is the combination of the on-board data and external environmental data. The training data for the energy consumption model is the combination of the data from vehicle dynamics data, on-board data and external environmental data.
6. **Validation of the models** was done by applying the models on the validation data to compare predicted values to the true energy consumption.
7. **Test on other trips** of the Zeekr dataset to determine the performance in real world conditions.

2

Theory

2.1 Vehicle Energy Consumption Model

2.1.1 Vehicle Dynamics Model

A vehicle moving at a constant velocity on a flat road encounters two main forms of resistance: rolling resistance from the ground and air resistance from the environment, denoted as F_f and F_w respectively. When the vehicle ascends a city road incline, it faces an additional force opposing its motion, known as gradient resistance, symbolized as F_i. Furthermore, the vehicle must counteract acceleration resistance, denoted by F_j, when accelerating. Consequently, the cumulative resistance experienced by the vehicle in motion is the sum of these components, expressed as (2.1):

$$F_{total} = F_f + F_w + F_i + F_j \tag{2.1}$$

The model is based on a basic physical model that describes the forces applied to the vehicle as it moves. The resistance that the wheels need to overcome when traveling can be written as(2.2):

$$F_{total} = mgf\cos\alpha + \frac{C_d A}{21.15}u_a^2 + mg\sin\alpha + (m + m_f)\frac{\mathrm{d}u}{\mathrm{d}t} \tag{2.2}$$

In the equation, the significance of each parameter is as follows

F_{total}: Total vehicle travel resistance [N]
m: Total vehicle mass [kg]
f: Vehicle rolling resistance coefficient
α: Road gradient angle [°]
C_d: Vehicle air resistance coefficient
A: Vehicle equivalent cross section [m^2]
u_a: Vehicle speed relative to wind speed [km/h]
m_f: Fictive mass of rolling inertia [kg]
du: Acceleration of the vehicle from point i to point j [m/s]
Thus the energy consumption of the resistance can be written as:

$$W_{res} = F_{total} \cdot v_a \cdot \Delta t \tag{2.3}$$

The consumption of resistance energy during vehicle travelling accounts for the majority of the total energy consumption [9]. Aerodynamic resistance and slope

resistance are the main contributions to vehicle resistance. The source of these two resistance components is related not only to the road conditions under which the vehicle is travelling but also to the design parameters of the vehicle itself. For the experiments in the report, we mainly used Zeekr 001 as our experimental data source vehicle and calculate the resistance of the vehicle.

2.1.2 Relevant Parameters

2.1.2.1 Vehicle Speed

Vehicle speed is an important factor in the EVs energy consumption. Generally, when the vehicle speed increases, the energy consumption goes up with it. In the dynamic equation (2.2) of driving resistance, it can be seen that the square of the relative speed of the vehicle is positively related to the resistance. This indicates that vehicle speed essentially affects energy consumption.

2.1.2.2 Vehicle Acceleration

Acceleration is a parameter that is constantly changing as the vehicle travels. There are various factors that affect acceleration, this includes driver behaviour, type of road etc. In addition, acceleration is also an important element in the vehicle dynamics equations(2.2) because the acceleration resistance is included in the equations. When acceleration increases, resistance also increases.

2.1.2.3 Slope

A greater slope leads to an increase in the energy consumption of an electric vehicle. This is because the vehicle needs more energy to overcome gravity, and in order to maintain the speed when travelling uphill, the EVs motor needs to produce more torque, which directly increases the energy output of the battery.

2.1.2.4 Ambient Temperature

The performance of batteries in electric vehicles is strongly affected by temperature. From Fig.2.1, it can be concluded that the effective capacity and energy output of the battery decreases in cold conditions. This means that the range of an electric vehicle is reduced in cold weather.

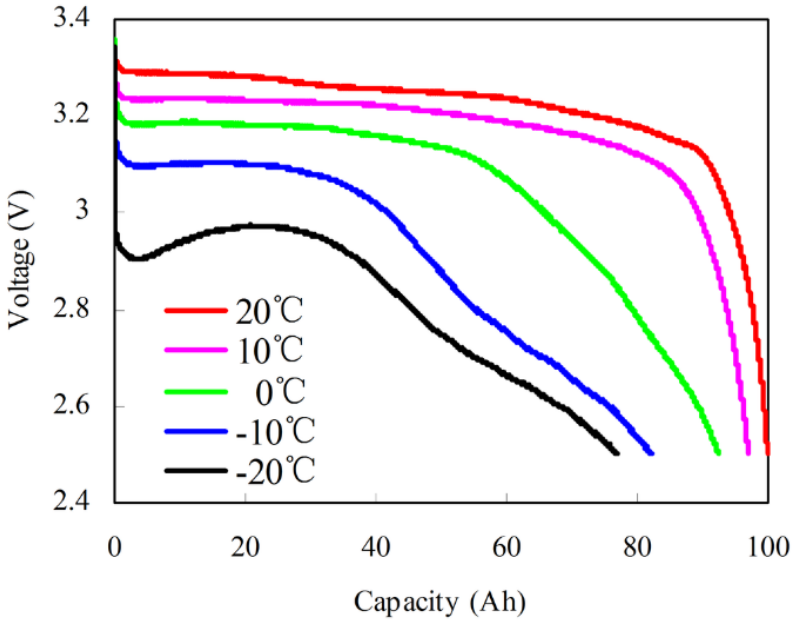


Figure 2.1: Discharge voltage curves at different temperatures [10]

The temperature in Sweden, the coldest months are January, February, with an average of -3.2 °C. The warmest month is July, with an average of 16.8 °C. The winter's duration also varies depending on area, from three months in the far south to nine months in Lapland. It can be seen that winter temperatures in Sweden are essentially below 0 °C and continue for long periods of time, so the influence on the batteries is very significant.

2.1.2.5 Wind Speed and Wind Direction

Wind speed and direction can significantly affect the energy consumption of an electric vehicle by influencing aerodynamic resistance. Headwinds (wind blowing directly in the direction of travel) increase the aerodynamic resistance on the vehicle. This means that the vehicle's motors have to work harder to maintain speed, which leads to higher energy consumption. Conversely, a tailwind (wind blowing in the same direction as the vehicle is travelling) reduces aerodynamic resistance. From equation 2.2, the effect of wind speed on aerodynamic resistance is square-scale, and this non-linear relationship makes whether upwind, downwind, or sidewind, have a particularly large impact on the energy consumption of electric vehicles. In addition, when the ambient temperature is low, the frequency of using air-conditioning in the vehicle increases, which also increases the energy consumption of the vehicle.

2.1.2.6 Road Category

The type of road also affects the energy consumption of EVs. On urban streets, EVs' speeds are low and there are many traffic signals, this stop-and-go driving behaviour allows for regenerative braking, which can put a lot of power back into the battery. In contrast, EVs consume more energy when driven at higher speeds

for longer distances on the highway. This is due to increased wind resistance when traveling faster and limited opportunities for regenerative braking.

2.2 Vehicle Speed Prediction Model

2.2.1 Relevant Parameters

In predicting vehicle speeds, it is important to take into account the effects of speed limits and traffic speeds, as well as the effects of road type.

2.2.1.1 Speed Limit and Traffic Speed

Speed limit and traffic speed represent information about the traffic flow on the road. In heavy traffic or congested areas, drivers often experience stop-and-go conditions. This frequent starting and stopping allows for the recapture of energy through regenerative braking systems, replenishing the battery charge. Although frequent acceleration requires more energy, the effect of regenerative braking often offsets some of this consumption.

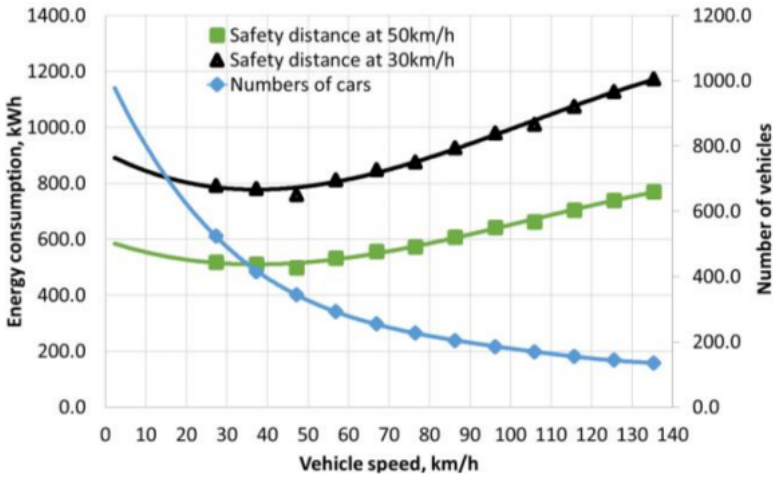


Figure 2.2: Dependency of energy consumption for a fleet of electric vehicles at different speed limit [11]

Different parts of the road have different speed limits and traffic flows, all of which affect the energy consumption of the EVs. From Fig.2.2, the reduction in energy consumption is due to the decreasing number of vehicles in the fleet along with the increasing speed on a given road section. So, both speed limits and traffic speed influence an EV's energy consumption by affecting the vehicle's speed, and traffic patterns.

2.3 Machine Learning Model

Regression analysis has been the most popular modeling technique in predicting energy consumption [12]. Machine learning is a good way to perform regression analysis. This section will give an introduction about all the machine learning model we used.

2.3.1 K-Means

Clustering is a unsupervised learning method that groups similar data samples into sets, known as clusters. K-means is a partition-based clustering method. The principle is to first initialize cluster centers, then assign each sample to the cluster whose center is nearest based on the distance between the sample and the center.

K-means is a partition-based clustering method. First initialize k cluster centers. Second assign each sample to the cluster whose center is nearest based on the distance between the sample and the center. Third recalculate its cluster class center position for each cluster class. Repeat steps 2 and 3 above until the centers of the clusters remain unchanged. The algorithm iterates to minimize the distance between samples and their respective cluster centers.

2.3.2 Linear Regression

Linear regression is used for modeling the relationship between a dependent variable (target) and one or more independent variables (features). The goal is to find a linear equation 2.4 that best fits the data, which can be used to predict values of the dependent variable.

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 + ... + \epsilon \tag{2.4}$$

y is the target value. $x_1, x_2, ...$ are the features. $\beta_1, \beta_2, ...$ are the weights. β_0 is the y-intercept. ϵ is the error term. Linear regression assumes that there is a linear relationship between the target and features. The errors are normally distributed, and have constant variance.

2.3.3 Decision Tree

Decision tree has a tree-like structure. Breiman et al. devided dicision tree into classification tree and regression tree [13]. Regression tree structure comprises nodes arranged hierarchically, beginning with a root node and extending to bottom nodes, known as leaves. Nodes in the tree contain logical tests based on predictor variables, while leaves hold predictions made by the model. Fig.2.3 shows the structure of a decision tree.

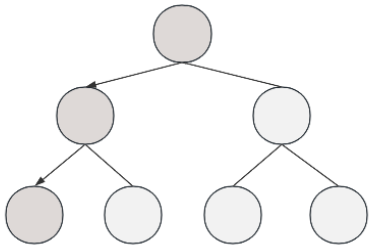


Figure 2.3: Decision Tree

Each path from the root to a leaf represents a conjunction of logical tests on predictors, delineating localized regions of the regression surface. It recursively partitions the data into different regions, with the target value in each region represented by the average of the samples in that region. Decision trees are suitable for capturing non-linear relationships, but they can be prone to overfitting.

2.3.4 Random Forest

As the depth of the decision tree increases, the decision rules become more complex, and the model is more prone to overfitting. A random forest is an ensemble model consisting of multiple decision trees. In a random forest, each tree independently predicts the output. For regression tasks, the final prediction is made based on the averaging of the predictions of individual trees. The structure of a random forest is illustrated in Fig.2.4.

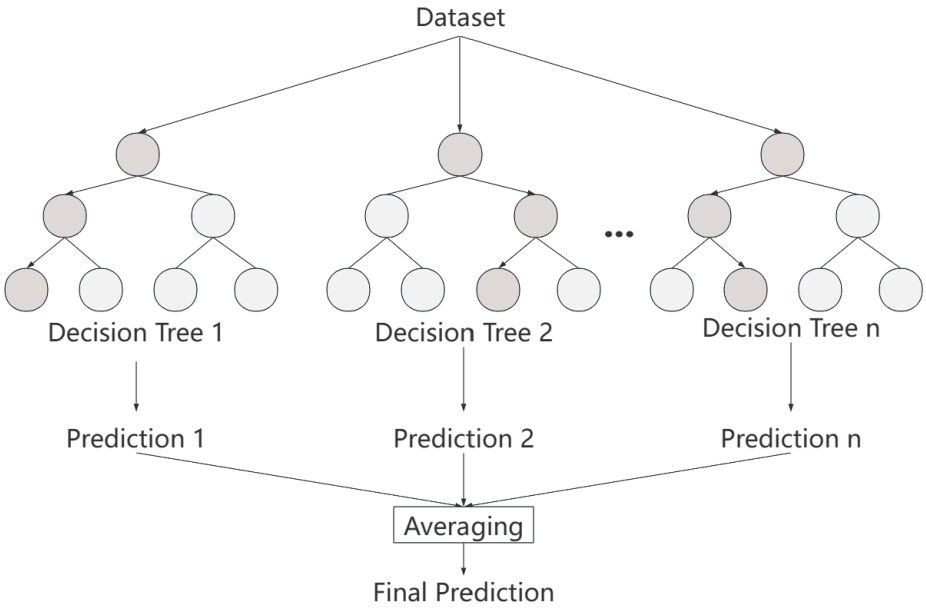


Figure 2.4: Random Forest

Random forest introduces diversity among trees by using bootstrapped samples and considering random subsets of features. Compared with decision tree, this approach helps random forest reduce overfitting.

2.3.5 K-Nearest Neighbor

K-Nearest Neighbors (KNN) is a supervised machine learning algorithm that can be used for both classification and regression tasks. It relies on calculating the distance between points in feature space to determine the nearest neighbors. Common distance metrics include Euclidean distance, Manhattan distance, and Minkowski distance. 'K' in KNN represents the number of nearest neighbors considered when making predictions. For regression tasks, prediction value is calculated by averaging the values of the K nearest neighbors, as shown in Equation 2.5.

$$f(x) = \frac{\sum_{i=1}^{K} y_i}{K} \tag{2.5}$$

KNN is classified as a "lazy learning" algorithm, which means it requires no training phase before making predictions. This can save time and computational resources. Besides, new data can be added without impacting the accuracy of the algorithm.

2.3.6 Ensemble Learning

Ensemble learning trains multiple models to solve the same problem, combining them to improve overall performance.Two popular methods of ensemble learning are voting and bagging. Fig.3.25 below illustrates the difference between voting and bagging.

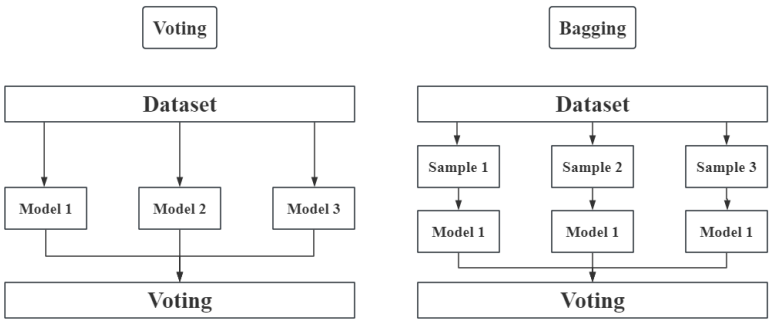


Figure 2.5: Voting vs Bagging

Voting involves combining the predictions from multiple machine learning models, which can be of different types. There are two main approaches: hard voting and soft voting. Hard voting determines the final prediction by the majority vote among the models. The final prediction of soft voting is made by averaging the predicted values from each model.

In contrast, bagging uses the same type of machine learning model trained on different subsets of the training dataset created through sampling. The final prediction in bagging is obtained by averaging the predictions. Random Forest in subsection 2.3.2 is a specific type of bagging method. It belongs to ensemble learning and builds multiple decision trees during training.

2.3.7 Long Short Term Memory

Long Short Term Memory (LSTM) is a type of recurrent neural network. In traditional RNN, the vanishing gradient problem can occur, blocking the network's ability to learn and cant keep information over long sequences. LSTM introduces a more complicated memory cell that can maintain information over extended periods. LSTM relieves the gradient vanishing and gradient explosion present in RNN networks. LSTM have been remarkably successful in various applications involving sequential data, such as time series prediction. The architecture of LSTM is shown as Fig.2.6.

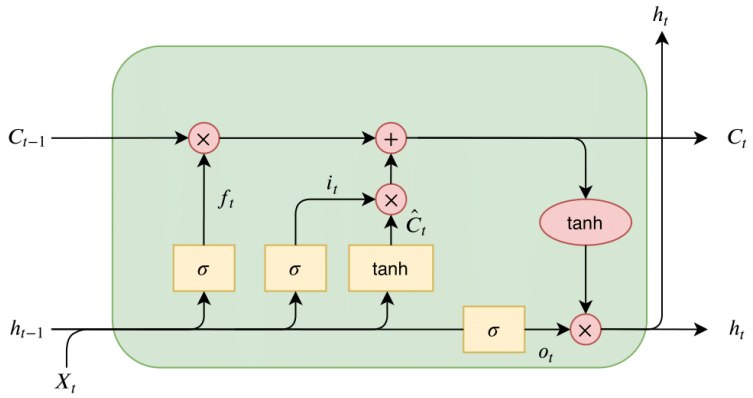


Figure 2.6: Long Short Term Memory

C, cell, is the "memory" of the network. It runs straight down the entire chain, with only a few linear interactions. LSTM has an essential component which is gates. LSTM has three gates to control the flow of information: the input gate, the forget gate, and the output gate. The input gate decides what information to let in. The forget gate controls what information should be discarded from the cell. The output gate decides what the next cell should be. These gates are composed of a sigmoid neural net layer and a pointwise multiplication operation.

2.3.8 Convolutional Neural Network

A typical Convolutional Neural Network (CNN) architecture, as shown in Fig.2.7, is formed by components of an input layer, two convolutional layers, and a fully connected layer.

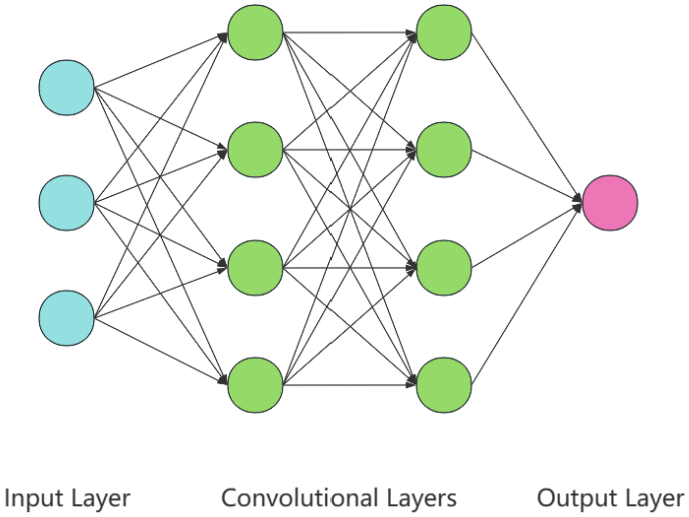


Figure 2.7: Convolutional Neural Network

The input layer takes in sequential data. The shape of the input includes sequence length and number of features. The convolutional layers consist of a set of filters, each of which can be seen as a stack of multiple convolutional kernels. These filters slide over the input data to do convolution operations to capture features. The weights on the filters are initialized with random values and are gradually optimized by learning from the training data. The fully connected layer is used to integrate features learned from previous layers for final predictions and produce the final output of the network.

A pooling layer can be added between the convolutional and fully connected layers. The pooling layer can effectively reduce the size of the parameter matrix, which reduces the number of parameters in the final connection layer, the pooling layer can speed up the computation and prevent the effect of overfitting.

2.3.9 Inputs and Outputs

In machine learning, the relationships between inputs and outputs are one to one, one to many, many to one, and many to many as shown in Fig.2.8. Each relationship represents a different way that data can be processed and predicted.

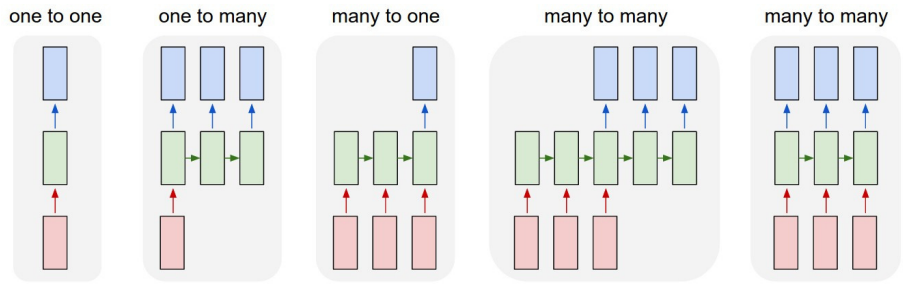


Figure 2.8: Relationships between inputs and outputs [14]

One to one is a single input mapped to a single output and commonly seen in tasks where a single data point produces a single result. One to many is a single input mapped to multiple outputs. This type is common in tasks like text generation where one input generates a series of outputs over time or steps. Many to one is multiple inputs mapped to a single output and is common in tasks where a sequence of data points is used to produce a single result. This type is used to do time series prediction because of using several past observations to predict a single future value.

Many to many is multiple inputs mapped to multiple outputs. There are two types of many to many relationships. They differ in the way they process and align these sequences. For the first many to many type, the entire input sequence is considered at once to generate the entire output sequence. This type is useful when the output sequence depends on the entire context of the input sequence rather than individual input. The second many to many type processes input sequence and output sequence in a strictly aligned manner, where each input corresponds to a specific output. This type is used when there is a direct mapping between the input and output sequences such as machine translation.

2.4 Statistical Analysis

This section primarily introduces tools for statistical data analysis. We use a correlation matrix to identify the parameters related to the energy consumption of electric vehicles. In addition, Variance Inflation Factor (VIF) is used to decide which features to include in our model. We use Kernel Density Estimation (KDE) to analyze the distribution of the input data. For the output data, we employ regression matrices to analyze data and inform more effective strategies for forecasting energy consumption in EV.

2.4.1 Correlation Matrix

The correlation matrix consists of the correlation coefficients between the columns. The element in the i-th row and j-th column of the correlation matrix represents the correlation coefficient between the i-th and j-th columns of the original matrix. Correlation matrix is symmetric, meaning the value at row i, column j is the same as the value at row j, column i. The diagonal elements are all 1 because each variable is perfectly correlated with itself.

V_i and V_j are the i-th and j-th columns of the original matrix. The correlation coefficient between the i-th and j-th columns $r_{V_iV_j}$ is calculated using the Pearson correlation coefficient formula [15].

$$r_{V_iV_j} = \frac{\sum(v_i - \bar{V_i})(v_j - \bar{V_j})}{\sqrt{\sum(v_i - \bar{V_i})^2 \sum(v_i - \bar{V_j})^2}} \tag{2.6}$$

v_i and v_j are the individual sample points of V_i and V_j. $\bar{V_i}$ and $\bar{V_j}$ are the mean values of V_i and V_j. The value of correlation coefficient is between -1 and 1. 1 indicates a

perfect positive correlation. -1 indicates a perfect negative correlation. 0 indicates no correlation.

2.4.2 Variance Inflation Factor

Variance Inflation Factor (VIF) helps detect the issue of multicollinearity. Multicollinearity occurs when two or more features are highly correlated, meaning they provide redundant information. High multicollinearity inflates the standard errors of the coefficient estimates. This makes it more difficult to determine the significance of individual predictors by correlation matrix. Multicollinearity can cause the estimated coefficients to be sensitive to changes, making the model unstable.

VIF quantifies how much the variance of a coefficient is inflated due to multicollinearity with other features. A high VIF indicates that the feature is highly collinear with one or more other features. Addressing high VIFs generally improves the quality of our regression model.

$x_1, x_2, ...$ are features. For each feature, we treat it as the dependent variable and regress it on all other features. The first feature can be expressed as equation 2.7.

$$x_1 = \beta 0 + \beta 1 x_1 + \beta 2 x_2 + ... + \epsilon_1 \tag{2.7}$$

Calculate the Coefficient of Determination R_1^2 obtained from this regression. Using the R_1^2 value, compute the VIF for X_1 using the formula 2.8.

$$VIF(x_1) = \frac{1}{1 - R_1^2} \tag{2.8}$$

VIF < 5 indicates low multicollinearity, which is generally acceptable. 5 VIF $<$ 10 suggests moderate multicollinearity and may be acceptable. VIF 10 signifies high multicollinearity, indicating that the feature is highly redundant and should be reconsidered.

2.4.3 Kernel Density Estimation

After acquiring the data, the first step is exploratory data analysis, which involves studying the distribution of the features (variables) and obtaining their distribution, to make the training set as comprehensive as possible.

The distribution density function of the data set is obtained in the following two ways. The first is the parameter estimation method, which assumes the data set fits a certain probability distribution. Then the parameters in the distribution are fitted according to the data set, e.g., likelihood estimation, Mixed Gaussian, etc. Since the parameter estimation method needs to incorporate subjective a posteriori knowledge, it is complicated to fit a model with the true distribution. The second method is nonparametric estimation, which does not include any a posteriori knowledge, but rather fits the distribution according to the features and properties of the data itself, which yields a better model than the parametric estimation method. Kernel density estimation is one type of non-parametric estimation [16].

Kernel density estimation (KDE) is an important nonparametric statistical method for estimating unknown probability density functions.KDE constructs a continuous probability density function by linearly summing discrete sample points to obtain a smooth distribution of samples.

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h\left(x - x_i\right) \quad = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{2.9}$$

K is the kernel function (non-negative, integrating to 1, consistent with the probability density property, and having a mean of 0), and h is a non-negative smoothing parameter, called the bandwidth. $K_h(x) = \frac{1}{h}K(\frac{x}{h})$ is the scaling kernel function.

KDE generates a continuous probability density function by smoothing the distribution of the data, which better reflects the true distribution of the data. Besides, KDE is good at identifying the multi-peak features of the data, and it can identify features such as concentration trends and outliers, which are used to identify anomalies or outliers. It plays an extremely important role in checking the data and balancing the dataset [17].

2.4.4 Regression Metrics

When assessing the performance of a regression model, three commonly used regression metrics are mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE).

2.4.4.1 MSE

MSE measures the average of the squares of the errors between the predicted and actual values.

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 \tag{2.10}$$

n is the number of samples, y_i is the actual value for the i-th sample, and $\hat{y}_i$ is the predicted value for the i-th sample. A lower MSE indicates a better prediction performance of the model to the data.

2.4.4.2 MAE

MAE measures the average of the absolute differences between the predicted and actual values.

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right| \tag{2.11}$$

n is the number of samples, y_i is the actual value for the i-th sample, and $\hat{y}_i$ is the predicted value for the i-th sample. A lower MAE indicates a better fit of the model to the data.

2.4.4.3 RMSE

RMSE is the square root of the average of the squares of the errors, which means the root of MSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{2.12}$$

n is the number of samples, y_i is the actual value for the i-th sample, and $\hat{y}_i$ is the predicted value for the i-th sample. Also, the lower the RMSE, the better fit of the model to the data.

3

Methods

3.1 Data Access and Processing

Both internal and external factors can affect electric vehicle energy consumption. These data are crucial for predicting energy consumption.

3.1.1 Internal Data

Internal data refers to information collected by the vehicle's internal systems and sensors that monitor and analyze various aspects of its operation. The data used in this thesis is from test vehicles with Zeekr, the test vehicles are mainly Zeekr 001. The data contains data from November 2023 to May 2024, so most of the data was collected in the winter. Due to the wide range of test vehicles driven, the data was split based on the total mileage driven to split the training set, i.e., 80% of the total range for the training set, 10% of the total range for the validation set, and 10% of the total range for the test set, and the details of how to split the dataset will be described in the following sections.

Figure 3.1: Zeekr 001 test vehicles

3.1.1.1 Data Access

Since the vehicle data is huge and has more than 140 features that can be selected, the first step is to choose and obtain the parameters related to the energy consumption of the vehicle, such as current, voltage, speed, and etc. The data is stored in a time-series database. As the vehicle data is recorded as 0 moments from start-up, Zeekr stores the data in a time series database and reset the time into local time for storage. This will help to distinguish the changes in vehicle energy consumption at different times of the day, and better construct the training set. For example, the energy consumption during the morning rush hour is different from the energy consumption in the afternoon when the traffic flow is low, and it is necessary to make sure that the data in the training set covers most of the time period when building the training set.

In Zeekr's Influxdb database, a filter has been set up to filter out data where the speed has remained constant at 0, since the vehicle may have only made starts and stops during these journeys and not travelled a distance. At the same time, we compared the data from the database with the logged data visualisation portal of the Zeekr test vehicle and filtered out the data that could be used.

The vehicle data was access by time series dataset. From influxdb we got the real time data of the vehicle such as latitude, longitude, direction, altitude and speed. For battery information we got the values of voltage and current of the battery. We acquired vehicle data for 200 trips. The total distance of 200 trips reached 7,214,896 meters. Most of these 200 trips are in Sweden and a few are in other European countries such as Norway and Spain.

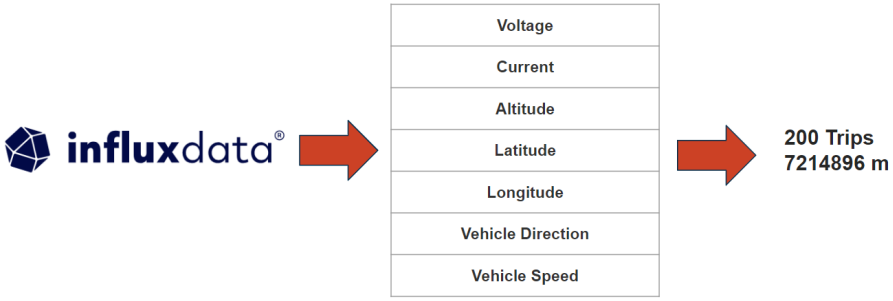


Figure 3.2: Vehicle data process

The data in Influxdb is more accurate since the raw data is preprocessed. The data is filtered first out vehicles in long stops when saving the raw data to Influxdb, and filters certain signals to filter out noise from certain signals, all of which makes the dataset cleaner and more accurate.

3.1.1.2 Data Synchronization and Interpolation

The data faces the issue of different sampling frequencies. To generate the dataset, the update frequency of GPS is 1HZ, and most of the other signals are updated at

100HZ. For example, the sampling frequency of the vehicles latitude and longitude data is lower than the frequency of the battery current and voltage, which means that between two neighboring GPS data there will be more current and voltage data. Therefore, we take the method of up-sampling and linear interpolation to complement the GPS data, as shown in Fig.3.3. So as to make sure that there is a GPS data corresponding to each current and voltage value, and avoid the loss of information of the dataset.

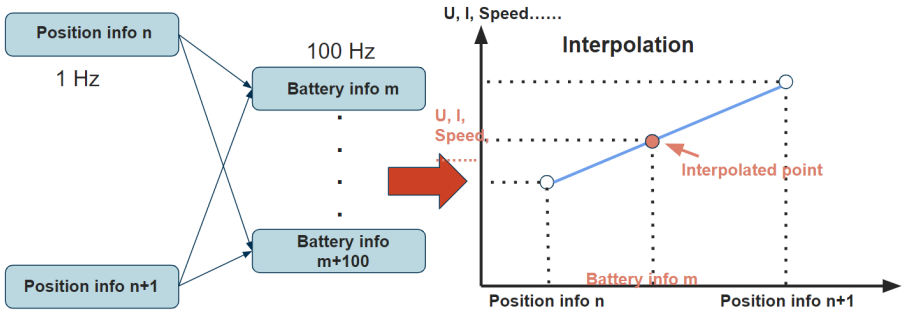


Figure 3.3: Vehicle data synchronization

Fig.3.4 shows an example of how the time-series data from the vehicle's GPS latitude data looks before and after interpolation. The data is taken from a 200-second short segment of the trip and the latitude signal is shown. The blue line shows the latitude without interpolation and the red line shows the latitude after interpolation.

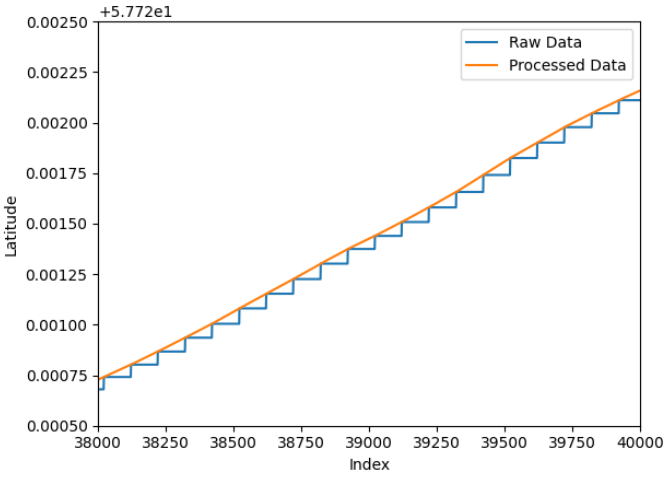


Figure 3.4: GPS data interpolation

3.1.1.3 Internal Data Processing

Fig.3.5 is our framework for vehicle data processing. We obtained the current and voltage values of the battery of the electric vehicle and latitude, longitude and altitude of vehicle's position. Then we calculated the energy consumption of the vehicle during driving, which was also our target value and output.

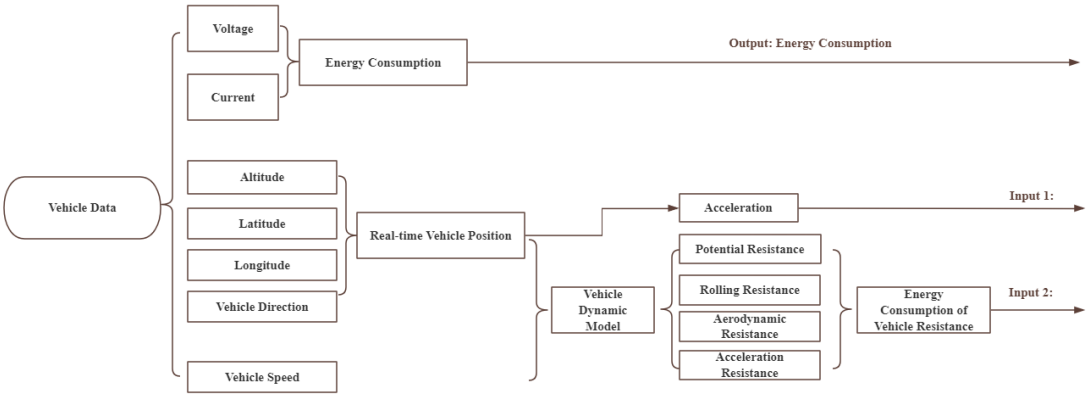


Figure 3.5: Vehicle data processing

Using the real-time position information of the vehicle, we calculated the acceleration of the vehicle, the energy consumption of the resistance, which was used as our feature and as inputs to the energy consumption prediction model.

3.1.2 External Data

External data refers to data originating from outside the vehicle itself that can impact energy consumption. Our external data includes weather data, traffic data and road data.

This data is obtained from the APIs of various suppliers and the APIs are inquired to get the real time data. In order to ensure the accuracy of the data, historical data is used as the data in the training set for the weather, wind speed, wind direction and other features. Since the traffic speed in map is the predicted value from the supplier and its accuracy cannot be identified, traffic speed will not be used as our feature in our speed prediction model. In addition, due to the limitation of the number of accesses to the API, downsampling is used to acquire the data during data collection. In Section 3.1.1.3, considering that the update frequency of vehicle GPS is much lower than that of other signals, we obtain API information based on the update frequency of GPS.

3.1.2.1 Weather Data Access and Processing

We got weather data from weather API. We sent a request containing time and vehicle position including longitude and latitude to the API. Then the API responded with data on temperature, wind speed, and wind direction. Weather data processing is shown as Fig.3.6. Temperature, wind speed and wind direction were used as inputs for our model.

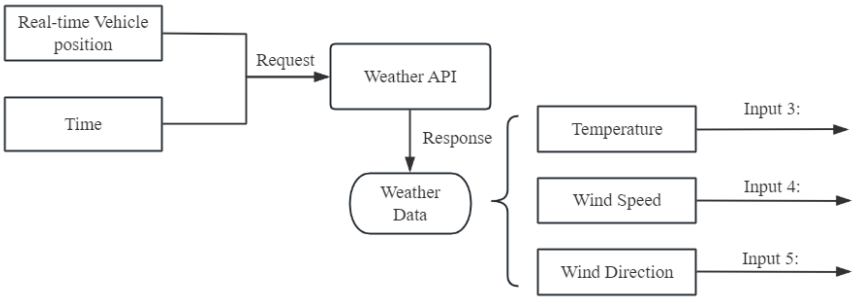


Figure 3.6: Weather data access and processing

3.1.2.2 Traffic Data Access and Processing

To access traffic data, we used previous-time vehicle position as departure, real-time vehicle position as destination and current time to send a request to traffic API. Then the API responded in JSON format. The response provided traffic speed and speed limit which were used as inputs of our model. The process is illustrated in Fig.3.7.

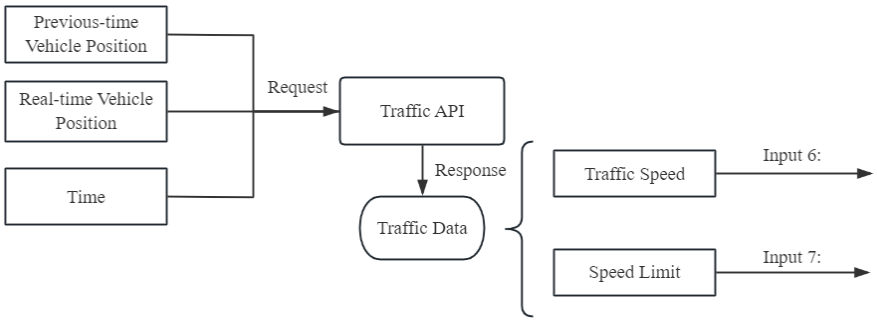


Figure 3.7: Traffic data access and processing

3.1.2.3 Road Data Access and Processing

We sent a request with real-time vehicle position data to the road API. The response from road API was in JSON format and included road type information as strings. These strings were converted to integers before being used as input for ML model training. These integer values were then used as input features. The process is shown in Fig.3.8.

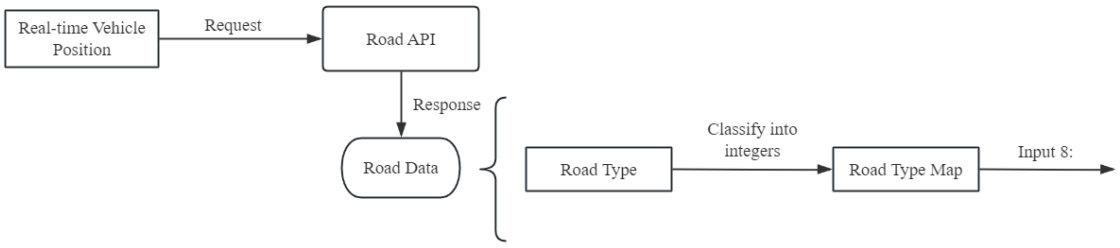


Figure 3.8: Road data access and processing

3.1.3 Vehicle Dynamic Model (VDM)

We use the vehicle dynamics model mentioned in Section 2.1 as our physics model. The model represents how vehicles consume energy by accounting for the resistance forces. It allows for precise calculations of how resistance factors affect energy consumption. In addition, the VDM also calculates some parameters that may be used, such as slope climb, aerodynamic factors and road curvature.

3.1.3.1 Resistance energy consumption

By calculating the energy consumption of the vehicle's resistance and comparing it with the total energy consumption of the vehicle during driving, as shown in Fig. 3.9, we find that the energy consumption of driving resistance accounts for 83% of the total energy consumption, which means that the energy consumption of driving resistance is an important part of the total energy consumption. To represent the consumption of the auxiliary, the formula was then extended with a scaled term. The simplified linear representation of the energy consumption of the EV can now be written as:

$$E_{total} = E_{resistances} + E_{auxiliary} = \alpha E_{resistances} \tag{3.1}$$

In the equation, $E_{Auxiliary}$ refers to the electrical energy consumption of different electrical components, such as air conditions, etc., and α refers to the regression coefficient.

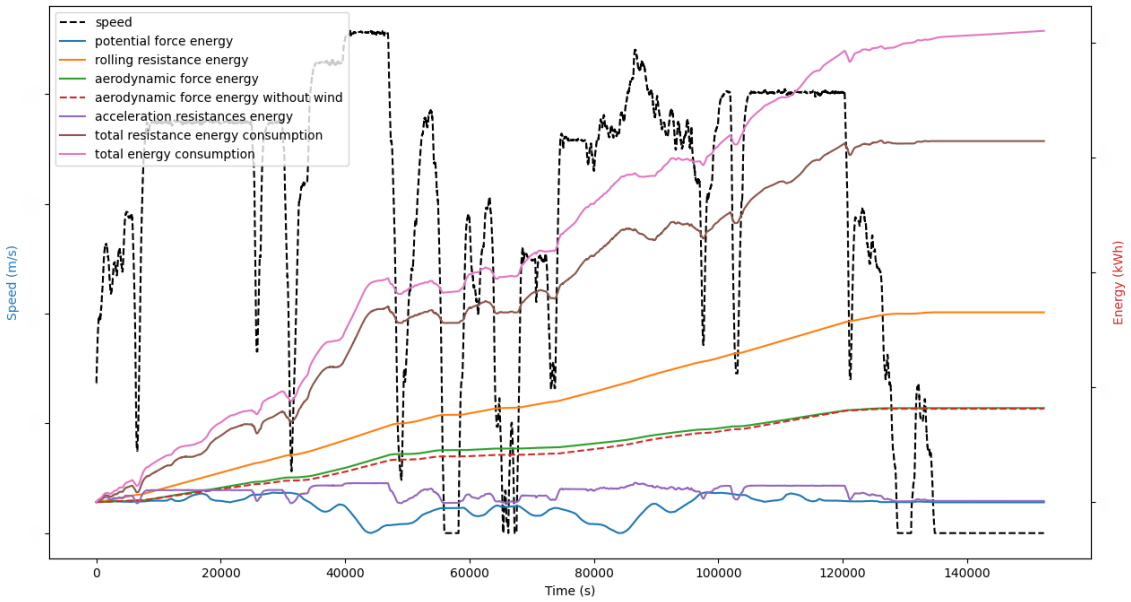


Figure 3.9: Contribution of different elements to the resistance of a vehicle in traveling

From Fig. 3.9, it can be seen that the main components of vehicle resistance are rolling resistance and aerodynamic resistance. From equation 2.2, we can get that the rolling resistance is related to the design of the vehicle, the road surface, and environmental factors. The aerodynamic resistance is related to vehicle speed, wind

speed and crosswind area. Besides, the speed is the velocity of the vehicle relative to the wind speed in aerodynamic resistance. However, the direction of wind speed is not always along or against the vehicle's direction of travel. In this thesis, we only consider the case where the wind speed and the vehicle's direction of travel are coextensive, so it is necessary to orthogonal decompose the direction of the wind speed to obtain the relative velocity by vector summation, as shown in Fig.3.10.

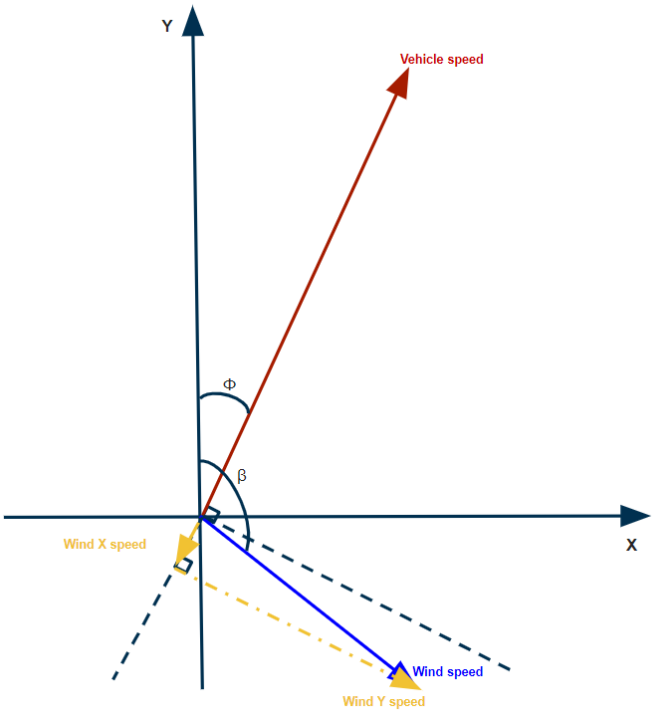


Figure 3.10: Speed orthogonal decomposition

In the figure, ϕ means vehicle heading angle and β means wind direction. Therefore, the equation for calculating the relative speed can be written as

$$V_{relative} = V_{vehicle} - V_{windX\,speed} = V_{vehicle} - V_{windspeed} \cos\left(180° - \beta + \phi\right) \tag{3.2}$$

To standardize the positive direction of the angle, we use clockwise as positive and scale the angle from 0° to 360°. Therefore, based on equation 3.2, 2.2 and 2.3, the resistance energy consumption of the vehicle during traveling can be calculated.

3.1.3.2 Slope Climbing, Aerodynamic Factor and Road Curves

Gradient resistance is an important component of vehicle resistance, so we take the change in slope as a separate feature, which is calculated as: Assume two points have the following coordinates and elevations: (ϕ_1, λ_1, h_1) and (ϕ_2, λ_2, h_2), ϕ and λ mean longitude and latitude h means altitude Then, calculate the horizontal distance d

between the two points using the Haversine formula:

$$\begin{aligned} R &= 6371000 \text{ meters (Earth's radius)} \\ \Delta\phi &= \phi_2 - \phi_1 \\ \Delta\lambda &= \lambda_2 - \lambda_1 \\ \Delta h &= h_2 - h_1 \\ a &= \sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_1) \cdot \cos(\phi_2) \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right) \\ c &= 2 \cdot \arctan 2(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned} \tag{3.3}$$

The equation for slope can be written as:

$$\Phi(i) = \frac{\Delta h_i}{\Delta d_i} = \frac{h_{i+1} - h_i}{d\left(\phi_{i+1}, \lambda_{i+1}, \phi_i, \lambda_i\right)} \tag{3.4}$$

where $d\left(\phi_{i+1}, \lambda_{i+1}, \phi_i, \lambda_i\right)$ is the horizontal distance calculated using the Haversine formula.

Aerodynamic factor (AF), shown in 3.8, is a translation of the speed profile for this method and represent respectively the performed accelerations and driving speed, v_{EV} means vehicle speed and v_w means the wind speed.

$$AF_i = \left(v_{EV_i} + v_{wi}\right)^2 \tag{3.5}$$

For the road curves, we also use Haversine formula and the equations are as follows:

$$d = 2r \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta \text{ lat}}{2}\right) + \cos(\text{ lat }_1) \cdot \cos(\text{ lat }_2) \cdot \sin^2\left(\frac{\Delta \text{ long}}{2}\right)}\right) \tag{3.6}$$

Where:
d: The distance between the two points (along the surface of the sphere)
r: The radius of the Earth
$lat_1, long_1$: The latitude and longitude of the first point
$lat_2, long_2$: The latitude and longitude of the second point
Δlat: $\Delta lat = lat_2 - lat_1$
$\Delta long$: $\Delta long = long_2 - long_1$

3.1.3.3 Real Energy Consumption

After data processing, all data are updated at a frequency of 10 milliseconds. We calculated the energy consumption due to driving resistance for each time stamp. We used current and voltage to compute the actual energy consumption of the vehicles at each update frequency through equation 3.7.

$$E_t = U_t I_t \Delta t \tag{3.7}$$

In the equation 3.7, E_t means this time point instant energy consumption, U_t and I_t mean instant voltage and current.In conclusion, through the vehicle dynamics model, we can calculate the resistance energy consumption of the vehicle during the driving, and we can calculate the relative speed of the vehicle and the angle of the wind direction relative to the direction of vehicle driving.

3.1.4 Data Extraction

All internal data are time series data, which we obtain from Influxdb. and save to CSV files. Since our data is larger and it takes more time to fetch the data from Influxdb each time, we take the data for each trip we fetched and save it in a CSV file to build a local database.

In the CSV file, each row represents the data at each point in time and each column represents the value of each feature value.In order to be able to predict between two places, we calculate the distance between two sampling points by speed and time of the vehicle, and then downsample at a frequency of per meter.

All the internal parameters, external parameters and parameters calculated through the VDM model for each trip are stored in separate CSV files, and the detailed description of each feature in the file is shown in Table 3.1.

Table 3.1: Implications of the features

Features	**Interpretations**
Time	Absolute sampling time
Latitude	Vehicle GPS real-time latitude position (°)
Longitude	Vehicle GPS real-time longitude position (°)
Altitude	Vehicle GPS real-time altitude (m)
Direction	Vehicle GPS heading from True North (°)
Speed	Vehicle real-time speed (m/s)
Acceleration	Vehicle real-time acceleration (m/s^2)
U	Vehicle real-time voltage (V)
I	Vehicle real-time current (A)
Temperature at 2m	Air temperature at 2 meters above ground (°)
Precipitation	Total precipitation (rain, showers, snow) sum of the preceding hour (mm (inch))
Rain	Rain from large scale weather systems of the preceding hour in millimeter (mm (inch))
Snowfall	Snowfall amount of the preceding hour in centimeters (cm (inch))
Snow depth	Snow depth on the ground (meter)
Wind speed at 10m	Wind speed at 10 meters above ground (m/s)
Wind direction at 10m	Wind direction relative to the direction of vehicle at 10 m above ground level (°)
Speed limit	Speed limit on a road (m/s)
Traffic speed	The rate at which vehicles travel on a roadway (m/s)
Week	Week of the year
Day	Day of the week
Hour	Hour of the day
Encode time	Encoding minutes and hours
Resistances	Vehicle driving resistance
Instant resistance energy consumption	EV Resistance energy consumption between two time points (kwh)
Total resistance energy consumption	Total EV resistance energy consumption at this moment (kwh)
Instant energy consumption	EV energy consumption between two time points (kwh)
Total energy consumption	EV total energy consumption at this moment (kwh)
Slope	Vehicle gradient (tanh)
Curve	Road curvature radius (km)
Category	Rough classification of road types
Type	Precise classification of road types
AF	Real-time aerodynamic factor of vehicle
Distance	Total distance at this moment (m)

Considering that the total distance travelled is usually time-relative, dividing by total distance can preserve the time-series characteristics of the data and ensure that the data used in model training and testing reflect the true time series of vehicle travelling. In real-world, predictive models are often based on historical data to predict future conditions. Dividing the training and testing data by total range better simulates this real-world situation and ensures that the model performs more reliably in the face of new data. Therefore, when dividing the dataset into training and test sets, the division is based on the total distance travelled (i.e., 80% of the data is used for training, 10% for validation, and 10% for testing) keeps the each full trip, as shown in Fig.3.11.

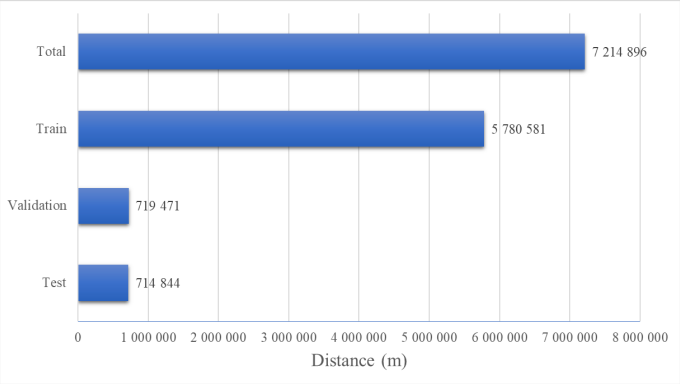


Figure 3.11: Dataset segmentation

3.2 Feature Selection

3.2.1 Correlation Matrix

In data analysis, correlation matrix is a commonly used tool to analyse and demonstrate the strength and direction of linear relationships between different features. Correlation matrices can help identify highly correlated features in data and provide a better way to understand how features in the data set are related to each other.

When more features are selected, it faces the problem of multicollinearity. In the correlation matrix, multicollinearity is usually expressed as a high degree of correlation between two or more features. Each element in the correlation matrix represents the correlation coefficient between the corresponding row and column features. The value of the correlation coefficient ranges between -1 and 1, with values close to 1 or -1 indicating strong positive or strong negative correlation, while values close to 0 indicate virtually no linear relationship. If the correlation coefficient of two features is very close to 1 (for instance, above 0.8) or very close to -1 (for instance, below -0.8), this indicates that there is a strong linear relationship between the two variables. This is an indication of a multicollinearity problem. Therefore, it is also important to avoid highly correlated features when selecting model features.

Fig.3.12 is the relevant parameters correlation matrix for speed prediction model. Among the many features, those with correlation coefficients higher than 0.05 were

selected as feature selection options for speed prediction. Fig.3.13 is the relevant parameters correlation matrix for energy consumption prediction model. Similarly, features with correlation coefficients higher than 0.01 are selected as feature selection options for energy consumption prediction. In the following sections, we will classify different inputs based on the correlation coefficient values under different thresholds and compare the prediction performance of the model under different inputs features.

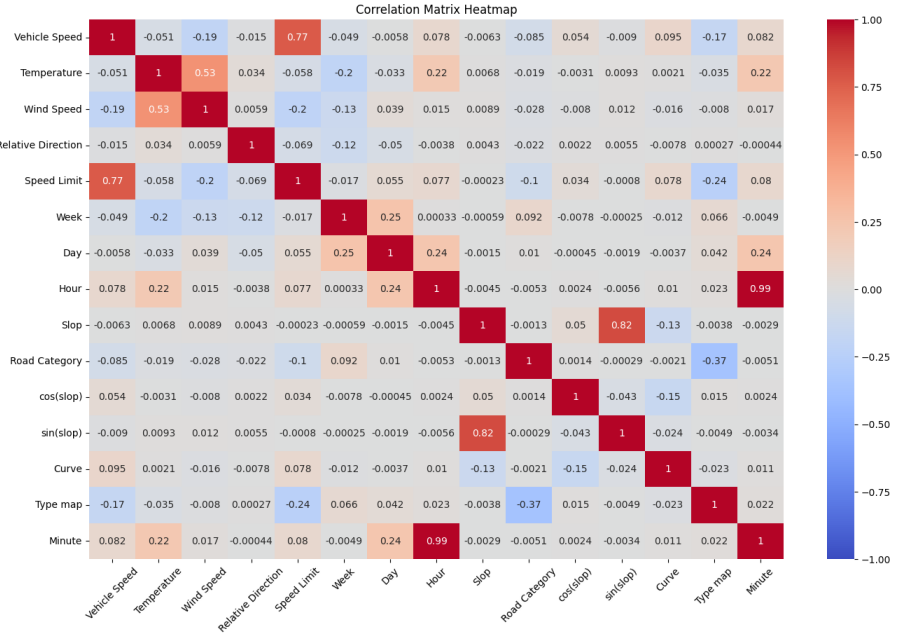


Figure 3.12: Speed prediction parameters correlation matrix

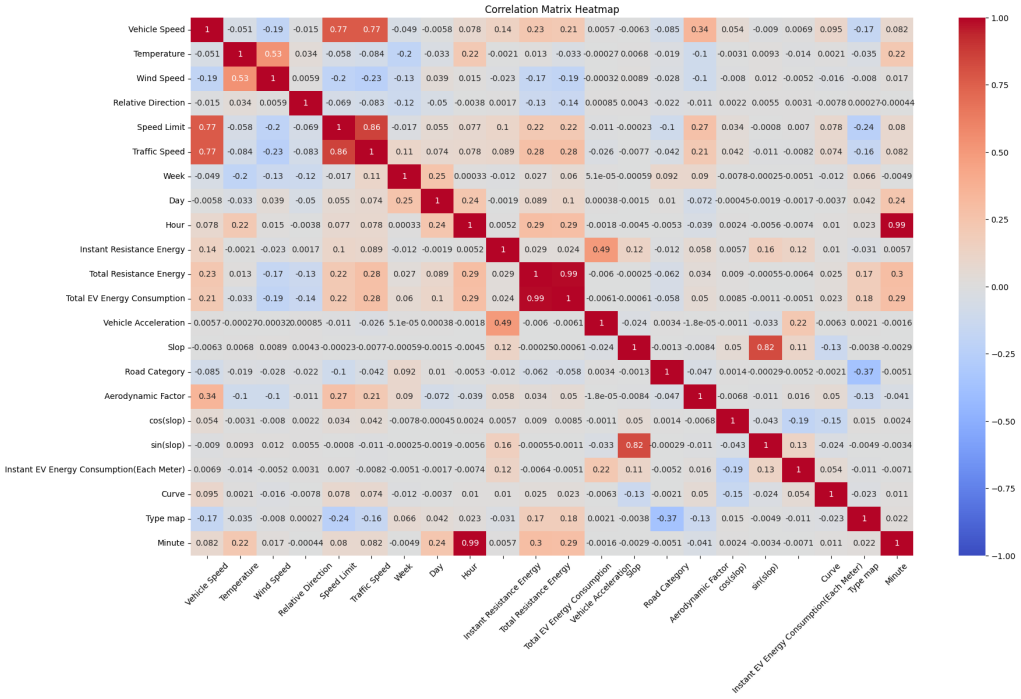


Figure 3.13: Energy consumption prediction parameters correlation matrix

3.2.2 Multicollinearity testing

Variance Inflation Factor is a commonly used statistical tool in multicollinearity detection. Multicollinearity refers to the presence of high correlation between predictor variables in a regression analysis, which leads to instability and difficulty in interpreting the regression coefficients, resulting in poor model predictions.VIF measures the correlation between a predictor variable and other predictor parameters.The higher the value of VIF, the stronger the correlation between the variable and the other predictor variables. Generally, if the VIF value of a variable is above 5, it is considered that there is severe multicollinearity [18].

After three rounds of VIF testing, the parameters with VIF values near 1 were selected as follows: Vehicle Speed, Temperature, Total Resistance Energy, Vehicle Acceleration and Slope, as shown in 3.2. It is important to note that a high VIF value does not always mean that the feature is not significant. Some features with high VIF values may have significant prediction ability for the model. Removing these features can cause the model to lose important information, which can reduce the prediction performance. In the Result section, we will compare the effect of different inputs on the model's prediction accuracy.

Feature	**VIF Selection 1**	**VIF Selection 2**	**VIF Selection 3**	**VIF Selection 4**
Vehicle Speed	33.974976	33.875821	9.571825	1.796179
Temperature	1.464429	1.415972	1.411082	1.001200
Wind Speed	4.669225	4.646357	4.583178	–
Relative Direction	3.559747	3.526627	–	–
Speed Limit	90.695557	86.171297	–	–
Traffic Speed	67.736825	67.452408	–	–
Total Resistance Energy	2.131566	2.017465	1.930659	1.795492
Vehicle Acceleration	1.003256	1.003150	1.000665	1.000652
Slope	1.017785	1.000948	1.000717	1.000656
Road Category	16.840124	14.215566	11.958614	–
Curve	9.704058	12.172644	–	–
Type map	14.247567	12.172644	10.914249	–
Encode time	21.491424	–	–	–

Table 3.2: Variance Inflation Factor (VIF) for different features

3.2.3 Features Final Section

By analyzing and comparing Fig.3.12, we remove the features which correlation coefficients lower than 0.05 in the heatmap of the correlation matrix, we can get the final feature selection for the speed prediction model.

For the instant energy consumption prediction model and total energy consumption prediction model, We classified two types of feature inputs based on the range of

correlation coefficients. The first category is correlation coefficients higher than 0.01, and we named this type of inputs as 13 feature inputs, as shown in Fig.3.5. The second category is the correlation coefficient higher than 0.02, and we name this category of inputs as 8 feature inputs, as shown in Table 4.6.

Table 3.3 shows the input parameters for speed prediction model, where minute is encoded for time, which will be explained in detail in the next section 3.4, and Table 3.4 and Table 3.5 show the input features for the instant energy consumption model and total energy consumption model.

Table 3.3: Speed Prediction Model Features Final Selection

Input	Speed limit
	Slope
	Type
	Category
	Minute
Output	Speed

Table 3.4: Instant Energy Consumption Prediction Model Features Final Selection

Input	Speed
	Temperature at 2m
	Wind speed at 10m
	Wind direction at 10m
	Traffic speed
	Speed limit
	Category
	Acceleration
	Instant resistance energy consumption
Output	Instant energy consumption

In Table 3.5, a number of choices are provided for the inputs of features to the energy prediction model, as when too many features are chosen as inputs, multicollinearity is unavoidable. However, if some features are deleted by VIF detection, the model may have the bad performance due to the lack of sufficient information to capture the complexity of the data. In the Results Section, we will compare the difference between multi-feature inputs and fewer feature inputs in detail.

3.3 Data Analysis

Kernel Density Estimation (KDE) is particularly important in data analysis as it provides a non-parametric method to estimate the probability density function of a dataset.

When separating the training and test sets, it is crucial to ensure that the parameter data in both datasets are in-distribution. The parameters in the test set should mir-

Table 3.5: Total Energy Consumption Prediction Model Features Final Selection

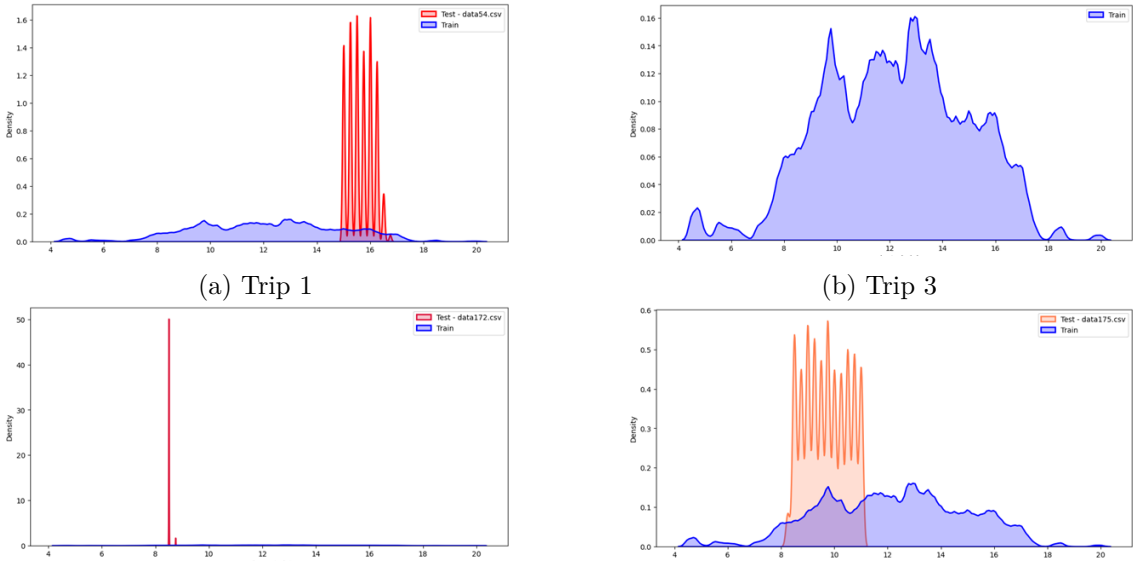
Input	Speed
	Temperature at 2m
	Wind speed at 10m
	Wind direction at 10m
	Traffic speed
	Speed limit
	Category
	Type
	Curve
	Week
	Day
	Encode time
	Total resistance energy consumption
Output	Total energy consumption

ror the distribution of the parameters in the training set to avoid out-of-distribution issues, ensuring consistency across both datasets. The generalization of the model can be compared by comparing the difference in KDEs between different test sets and training sets. In the next subsections 3.3.1, the KDE will be performed on the important features of the different models. We have included some of the figures from the data analyses in the following sections.

3.3.1 Speed Prediction Model Test Dataset Analysis

In the speed prediction model, time is an important feature in the model. At the same time, minute-level resolution is important because hourly resolution may not be sufficient to capture the effect of time of day on traffic speeds. This feature mainly indicates the distribution of the driver's travelling time, which we transformed as described in the following subsections.

From Fig.3.14, it can be concluded that the driving time of the training set is distributed from 4 am to 8 pm, and at 10 am to 3 pm is the peak time of vehicle usage. By analysing the driving time of the test set it can be concluded that the training set contains data that captures the data of the test set, which means that our dataset is better segmented (for example, as shown in Fig.3.14d). In some trips, for example trip 3, as shown in Fig.3.14b, it does not display its KDE distribution because the driving time is shorter which leads to the same value of minute.



(a) Trip 1

(b) Trip 3

(c) Trip 17

(d) Trip 18

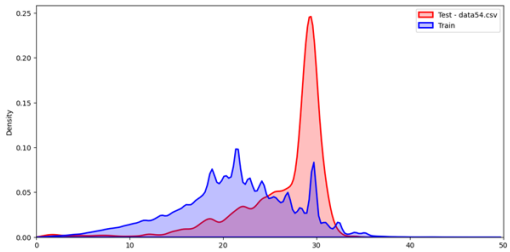
Figure 3.14: Time Test Dataset Analysis

3.3.2 Energy Consumption Prediction Models Test Dataset Analysis

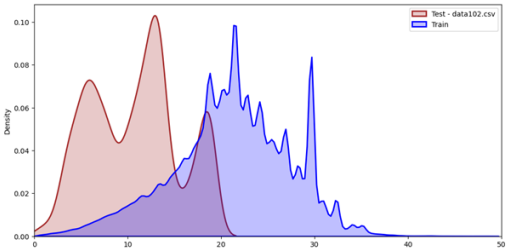
Energy consumption models include instant energy consumption prediction model and total energy consumption prediction model. In Subsection 2.1.2 Relevant Parameters, it can be learnt that vehicle speed and external environmental factors are particularly important for the performance of the model. Therefore, the KDE of vehicle speed, wind direction, wind speed, traffic speed and speed limit will be analysed.

Fig.3.15 shows the KDE distribution of speeds, and it can be concluded that the vehicle speeds in the training set are from 0 m/s to 40m/s, and some of them are over 45m/s. The training data shows a wider distribution of speeds, which indicate that it contains a wide range of different driving conditions or scenarios. In the test trips, for example trip 4 as shown in Fig.3.15b, which had significant concentrations at specific speed values in a dataset derived from a relatively more consistent driving environment. Models trained on the training data may perform poorly when faced with a unique driving condition similar to these trips.

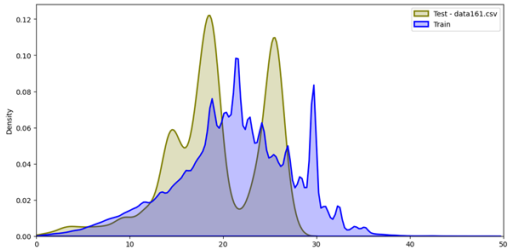
Fig.3.16 shows the distribution of wind direction data. The training data shows a wider distribution of wind direction. However, there is a peak in the north direction close to 0°/360° degrees, the overall distribution is more balanced, especially in the 100° to 300° range, compared to the training data which contains more eastern and western winds to the vehicle. In the test trips, the training set contains almost all of their wind direction situations.



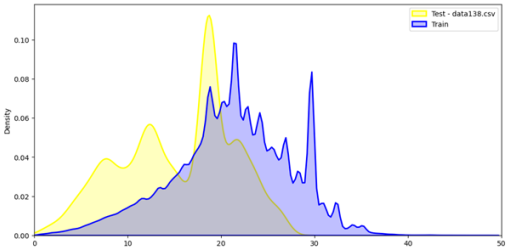
(a) Trip 1



(b) Trip 4

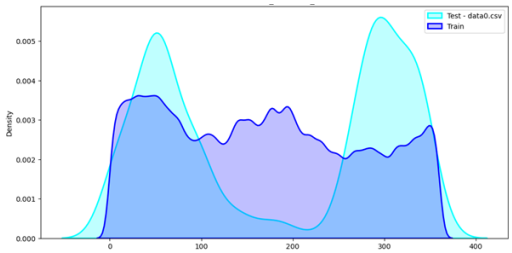


(c) Trip 7

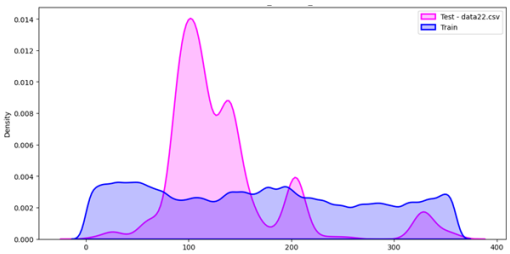


(d) Trip 11

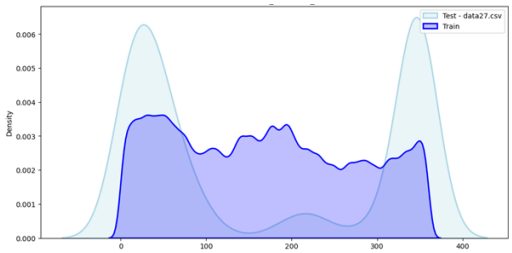
Figure 3.15: Vehicle Speed Test Dataset Analysis



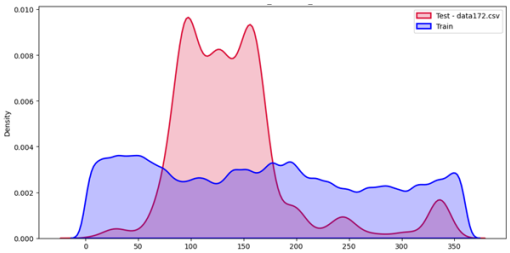
(a) Trip 1



(b) Trip 7



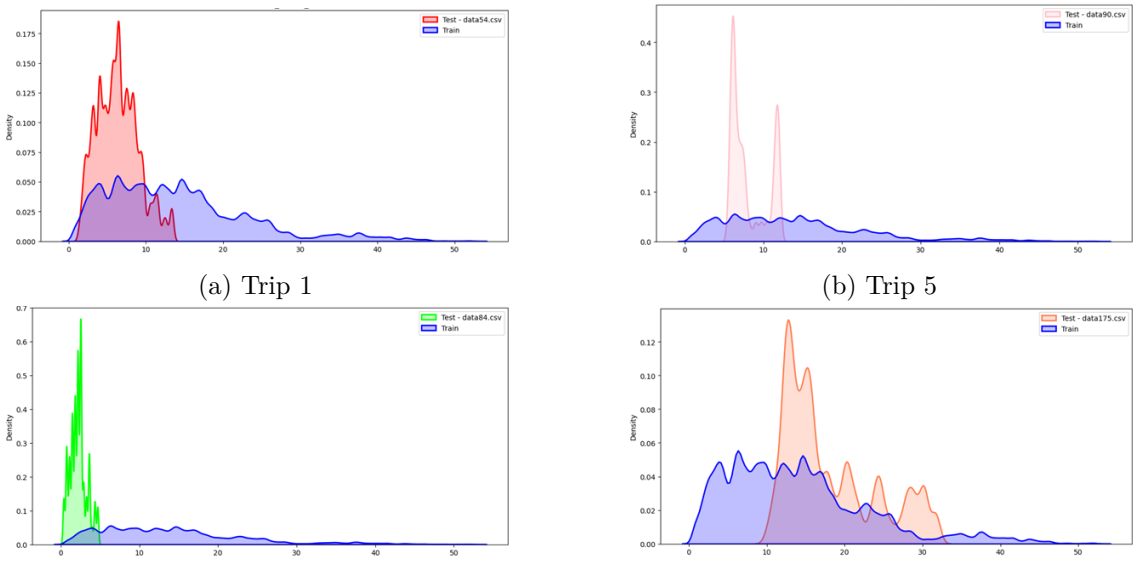
(c) Trip 9



(d) Trip 14

Figure 3.16: Wind Direction Test Dataset Analysis

From Fig.3.17, we can get that the wind speed distribution in the training set is smooth and broad, covering a wider range of wind speeds from nearly 0 m/s to more than 30 m/s overall. In the test trips, however, there is always a very obvious spike in their wind speed distribution, which indicates that the wind speeds in this dataset are mainly centered on this range, with fewer other wind speeds. This is because there were no sudden changes in the weather during the trip and the test data set is a more specific environment.

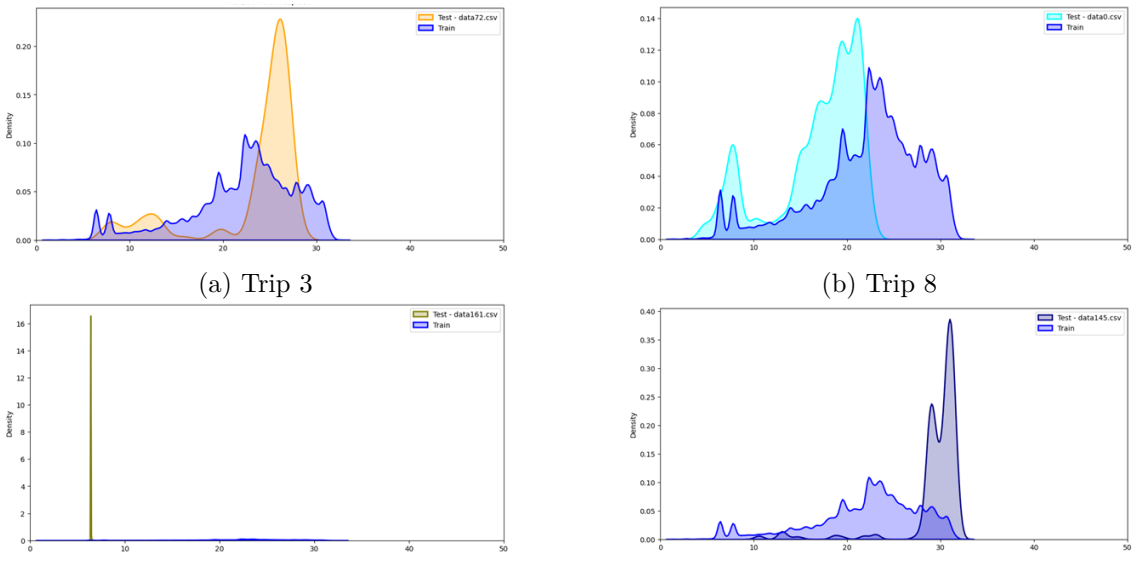


(a) Trip 1 (b) Trip 5

(c) Trip 13 (d) Trip 18

Figure 3.17: Wind Speed Test Dataset Analysis

Fig.3.18 shows a relatively wide distribution of the traffic speed training data, with high peaks around about 15 m/s and 25 m/s, indicating that the training data covers a wider range of speeds. However, in trips 7 (as shown in Fig.3.18c) and 15 (as shown in Fig.3.18d), it shows a very sharp and narrow peak almost at this position of 10 m/s. This sharp peak indicates that all the observations in the test data set are almost exclusively focused on this particular speed value, which may have resulted in the poor performance of the model in the tests for these two trips.



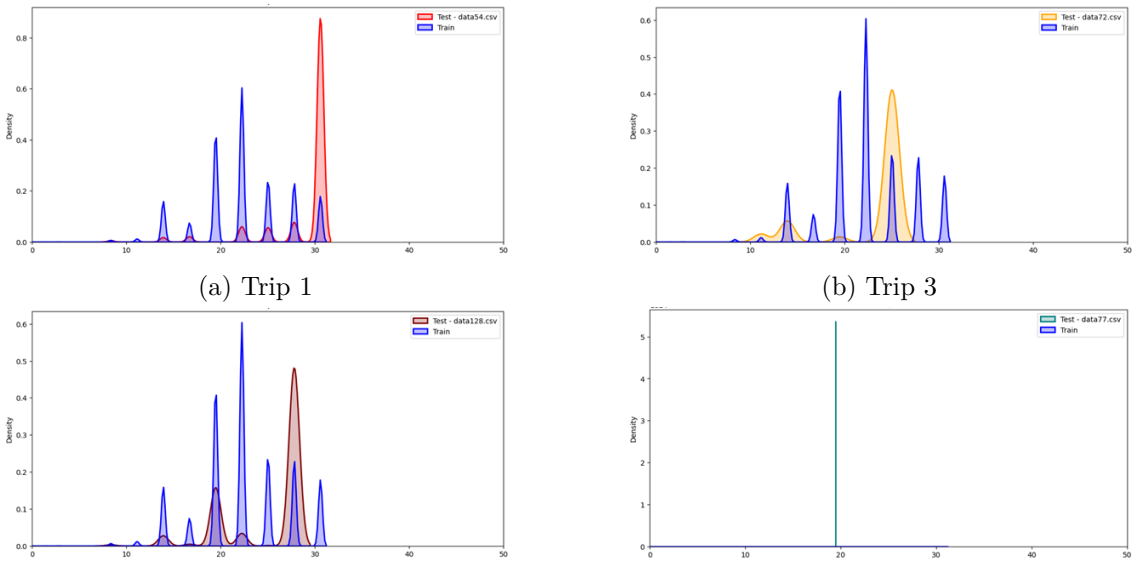
(a) Trip 3 (b) Trip 8

(c) Trip 7 (d) Trip 15

Figure 3.18: Traffic Speed Test Dataset Analysis

Fig.3.19 shows that the speed limit training data has a wider distribution and contains multiple peaks of speed limit values, with more significant peaks around 20m/s

and 30m/s. The speed limit training data has a wider distribution than the vehicle speed data. However, in the distribution of vehicle speed it can be concluded that the average speed of the vehicle in training dataset is greater than 30m/s, according to which it can be concluded that the driver has aggressive driving behaviour.



(a) Trip 1

(b) Trip 3

(c) Trip 13

(d) Trip 15

Figure 3.19: Speed Limit Test Dataset Analysis

In summary, in our training set, the dataset generally shows a wider distribution compared to the test dataset. This indicates that the training data covers a wide range of driving conditions, which suggests that the model is likely to account for a wide range of real-world driving scenarios.

3.4 Vehicle Speed Prediction Model

Before predicting energy consumption, we need to predict vehicle speed.

The choice to use Long Short-Term Memory Networks (LSTMs) instead of traditional machine learning models for predicting driver's speed is mainly due to the fact that LSTMs are able to effectively deal with the specific challenges associated with time-series data. A driver's speed is not only dependent on current road conditions, traffic conditions and driving behaviours, but also be influenced by the previous seconds or minutes of driving conditions. This dependency is an important feature of time-series data, and appropriate models are needed to capture this temporal dependency. In addition, the structure of LSTM enables it to efficiently learn and maintain long-term data dependencies through a gate control (forget, input, output gates) mechanism. The length of the time series of driving data may vary depending on the length of the trip, which requires the model to be able to flexibly handle input sequences of various lengths.The ability of LSTM to handle input sequences of arbitrary length provides greater flexibility to the model. In the sections 3.4.1, we will introduce our window size and sliding size.

From the above analysis it can be concluded that it is important to use journey times are as inputs to the model. In the acquired data, we divided the time into week, days, hours, minutes and seconds. Since most of the data were collected in the winter and spring from 2023 to 2024, the week's data does not cover a full year, therefore it is discarded.

We use the random forest regression prediction as a baseline and analyse the importance of the parameters, as shown in Fig. 3.20. It can be concluded that minute and hour have relatively higher weights in the prediction process, so we choose these two parameters as our inputs. It is rare for the speed of a vehicle to change rapidly in 1 second, so we discard the seconds as well.

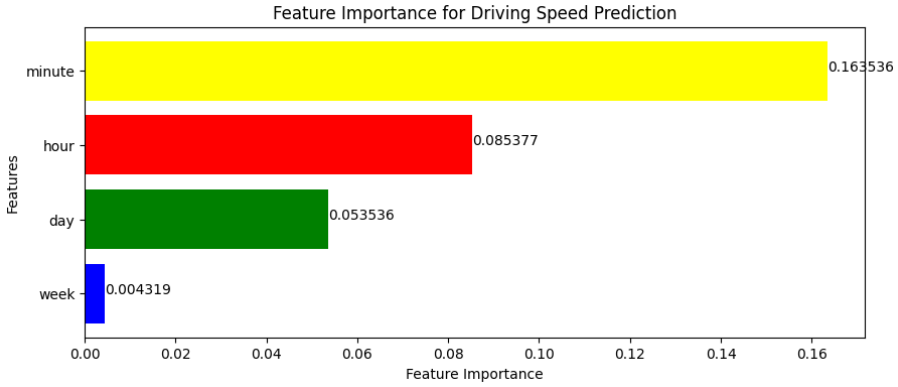


Figure 3.20: Feature Importance For Speed Prediction

Hours and minutes as discrete categorical variables that the model does not learn well from the variations in them, and so these two parameters need to be encoded. We encode the hours and minutes by combining them into one variable through a simple conversion. Treat the hours as an integer part and the minutes as a decimal part. Since traffic conditions are usually stable over short periods of time (e.g., 15 minutes), although speeds may change during manoeuvres such as overtaking, we chose to split the data into 15-minute time segments, as shown in algorithm 1.

Algorithm 1 Encode Time Algorithm

1: $minute \leftarrow$ current minute of the hour
2: $a \leftarrow \text{round}(minute/15)$
3: $encoded\_time \leftarrow hour + a \times \frac{1}{4}$

In the following subsections we explain how to slide the window to get the data and the construction of the model. By converting hours and minutes to a single continuous value, the model can more easily account for the linear flow of time.

3.4.1 Sliding Window Data Acquisition

Since we have a very large amount of data, we train the model using a sliding window approach to take values. The sliding window method can help the LSTM

learn and understand the time series dependencies in the data, and by choosing the appropriate window size, it can help the LSTM use its limited memory capacity more effectively to focus on the most recent information that is most important for prediction. When performing speed prediction, more training samples can be generated by slicing the time series data into multiple overlapping windows. This approach not only expands the dataset, but also maintains the structural properties of the time series, allowing the model to see the continuity of the data before and after in each time window. In the speed prediction model, set the window size to 50 and slide 20, as shown in Fig.3.21.

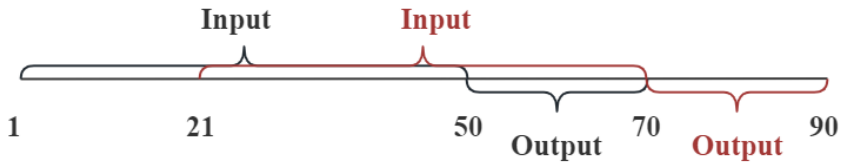


Figure 3.21: Speed model architecture (many to many architecture)

3.4.2 Model Architecture

We employed a neural network model, which adopts a sequential layer stacking approach. The architecture begins with a Long Short-Term Memory (LSTM) layer consisting of 128 units, utilizing scaled exponential linear units (SeLU) as the activation function. Following the LSTM layer, the model comprises a series of fully connected (Dense) layers with 64, 32, and 16 units respectively, each also using SeLU activation functions to facilitate nonlinear transformations and refinement of features extracted from the LSTM layer, as shown in Fig.3.26.

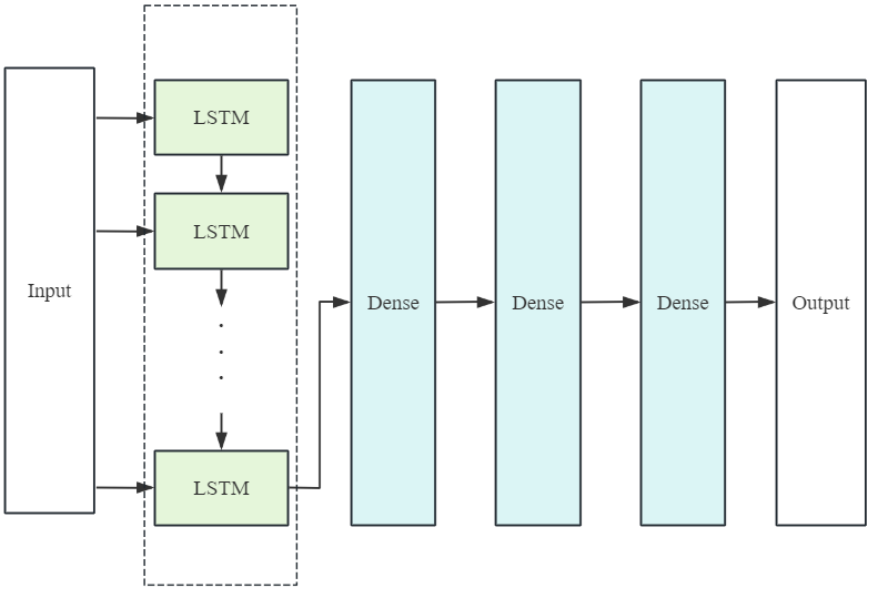


Figure 3.22: Speed model architecture

We tried two different LSTM frameworks, distributed as many to one and many to many. We will compare the performance of the two framewoin thes in Results Section.

3.4.2.1 Many To One

In the many-to-one structure, each input to the LSTM model is a sequence of length 50, and the window for each prediction slides 20 time steps over the time series. The window size is larger than the sliding time step is so that more historical data is recorded for the next prediction. The data from the first 49 time steps is used to predict the speed at the 49th time step. For example, given input steps [input0, input1, ...input49], the model should estimate the speed value at step 50, as shown in 3.23

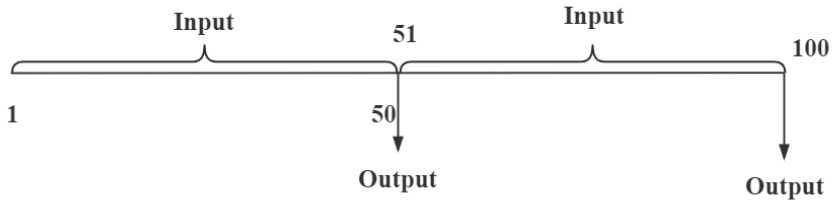


Figure 3.23: Many to on architecture

3.4.2.2 Many To Many

In a many-to-many architecture, the input to the LSTM model is a sequence of length 50 at a time, the window for each prediction slides 20 time steps up the time series, and the model output is the speed value for the next 20 time steps, as shown in 3.21.

3.5 Instant Energy Consumption Prediction Model

In the energy prediction model, we predict the total energy consumption of the vehicle. In the previous model, we converted all predicted and calculated values to international units, and energy was predicted in kWh.

3.5.1 Instant Energy

Instant energy refers to the amount of energy consumed by a vehicle over one meter. For example, the instant energy at distance d is the total energy at distance d+1 minus the total energy at distance d as shown in Fig.3.24. Then, sum up all the instant energy to get the total energy consumption.

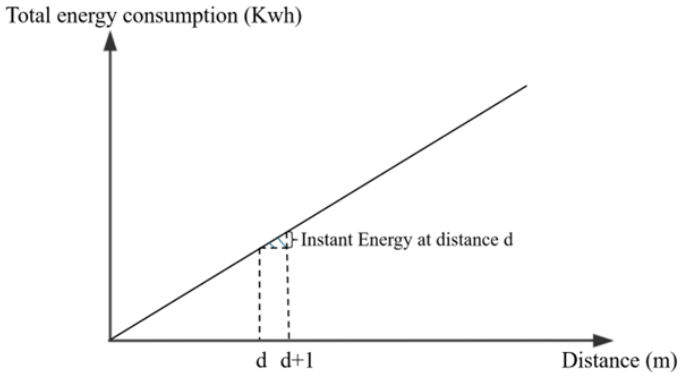


Figure 3.24: Instant Energy

3.5.2 Machine Learning Methods

When using machine learning for prediction, a large amount of data needs to be loaded for reading and writing and fed to the model for training, which requires a large amount of RAM space on the computer. Therefore, we need to downsample the data.

Therefore, we downsample the data using the Kmeans Cluster approach. After dividing the data into a number of classes through KMeans clustering, downsampling can effectively reduce the number of data points in each class. This method of compressing the data reduces the storage requirements. In addition, downsampling by clustering can reduce the number of data points evenly in each class, which helps to mitigate or avoid the excessive impact of certain classes on model training.

3.5.2.1 Basic Machine Learning Model

In equation 3.1 it can be concluded that it is a linear regression problem between the energy consumption of the vehicle and the other input parameters. Therefore, in the base machine learning model, we select Linear Regression, Decision Tree Regressor, K Neighbors Regressor and Randomforest Regressor as several choices and use these as the baseline. the inputs of these models are in the Table 3.4 show.

3.5.2.2 Ensemble Learning Model

Ensemble learning can combine the strengths of multiple base models, which makes the model predictions more accurate. We use the ensemble learning structure of voting, employing KNN, Random forest and Decision tree as base models, as shown in Fig.3.25. Integrated learning can reduce the risk of overfitting effectively. A single model may overfit the training data, but by integrating multiple different models, this problem can be reduced.

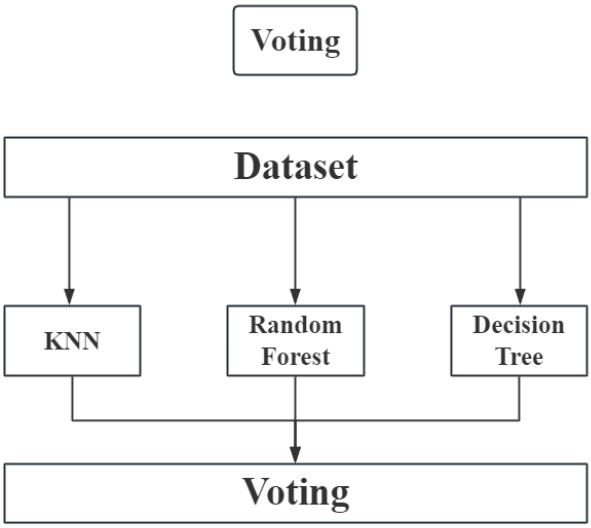


Figure 3.25: Ensemble learning model architecture

3.5.3 Deep Learning Methods

In the prediction of instant energy consumption, the data are not time series data, instead they are values of energy consumption at each moment, which are discrete. Therefore the models dealing with sequential problems are not suitable. So, the simple neural network model is built to make predictions. We added aerodynamic factor as an input and the equation is 3.8. AF is the speed profile translation of the method, which represents the executed acceleration and travelling resistance respectively.

$$AF_i = (v_{EVi} + v_{wi})^2 \tag{3.8}$$

In this model, in input (as shown in Table 3.4) we delete instant resistance energy consumption but add the aerodynamic factor, sin slope, cos slope, which is closely related to instant resistance energy consumption.

3.5.3.1 Model Architecture

The model framework consists of a series of alternating dense layers and dropout layers. It begins with a dense layer of 128 units, followed by a dropout layer to reduce overfitting. This is followed by another dense layer with 128 units and another dropout layer. Next, there is a dense layer with 64 units, followed by a dense layer with 32 units, and finally, an output layer with a single unit. The output of the last output layer is every moment instant energy consumption.

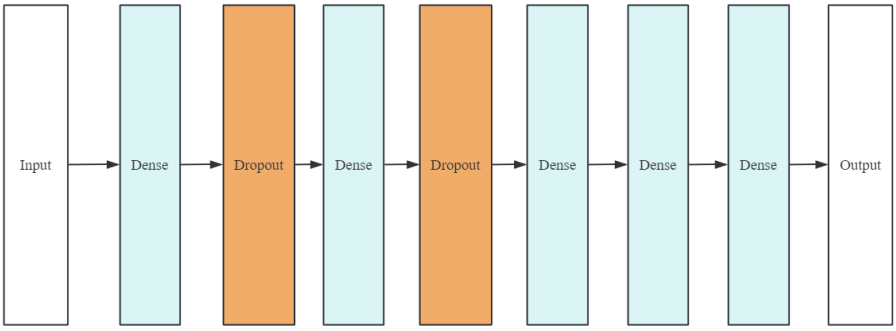


Figure 3.26: NN model architecture

3.6 Total Energy Consumption Prediction Model

Total energy consumption refers to the complete amount of energy used by a vehicle over a specific distance. It encompasses all the electricity energy used during the entire journey.

3.6.1 Model Architecture

Considering the excessive number of inputs to the model and the relative complexity of the data, we used CNNs+LSTMs to predict the total energy consumption. Combining CNNs and LSTMs leverages the strengths of both architectures. CNNs act as powerful feature extractors, while LSTMs provide the ability to understand temporal dependencies. This hybrid approach can lead to more robust predictions than using either architecture alone. Our architecture is shown in Fig.3.27

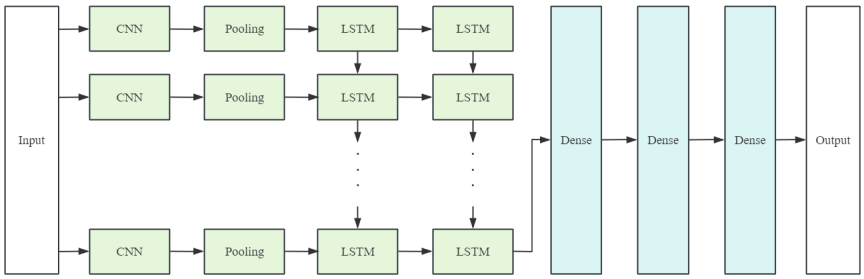


Figure 3.27: CNNs+LSTMs Architecture

We still tried the many-to-one and many-to-many model structures and finally used the error matrix such as MSE, MAE, RMSE to evaluate the performance of the models.

4

Results

4.1 Speed Prediction Model

Since the data of traffic speed comes from the predicted value of the supplier, there is no guarantee of its accuracy, so we divided the model into two types, one many to one model and one is many to many model. We used the traffic speed provided by the map API as a baseline. First, we calculate our prediction and traffic speed performance against the true speed. Then we compared our predictions to traffic speed.

4.1.1 Speed Prediction Model (Many to one)

The results of the error matrix of the speed prediction model under the model structure of many to one are shown in Table 4.1. It can be seen that the predicted value of the LSTM model we constructed using the structure of many to one is slightly better than the traffic speed of map,and its RMSE is 0.11 lower than the predicted result of map.

Table 4.1: Performance of speed prediction (Many to One)

Model	MSE	MAE	RMSE
Speed limit	15.4255	2.6381	3.9275
Traffic speed	15.4275	2.8642	3.9853
LSTM	14.9930	2.7663	3.8721

Fig.4.1 and Fig.4.2 shows the results of the comparison between the predicted and true values and the comparison between the predicted values and the predicted values provided by map for all the trips respectively.

From Fig.4.1, the orange prediction line generally follows the trend of the blue true line. This indicates that the model can capture the main patterns and trends in the data. However, it struggles with capturing the details and fluctuations accurately. The prediction line is smoother than the true line, suggesting the model has difficulty following rapid changes in speed.

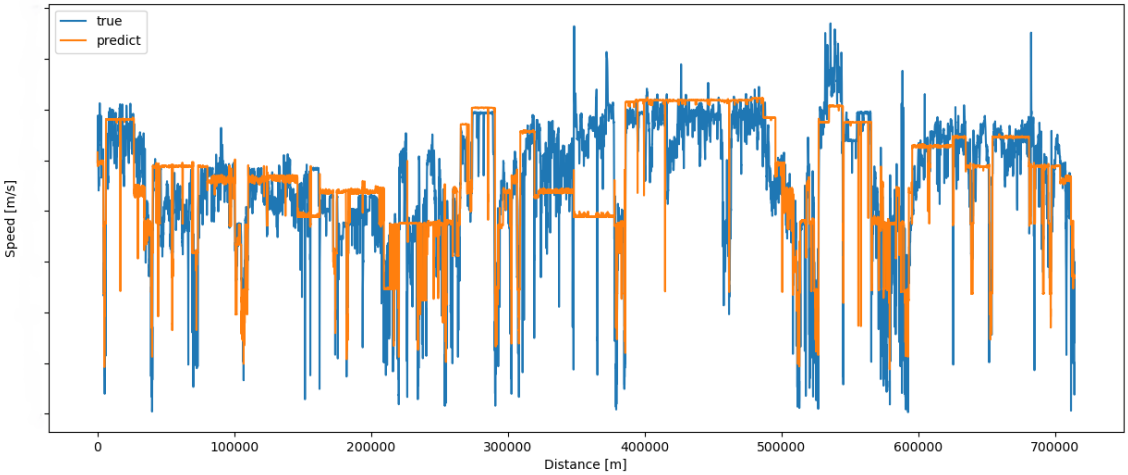


Figure 4.1: Speed prediction model many to one results

From Fig.4.2, we can find that the orange speed limit line is often higher than both the true speed and the predicted speed. This suggests that vehicles are frequently traveling below the speed limit, possibly due to traffic conditions or other constraints. In many regions, the predicted speed aligns closely with the traffic speed, this means that our prediction model is able to roughly predict the traffic speed.

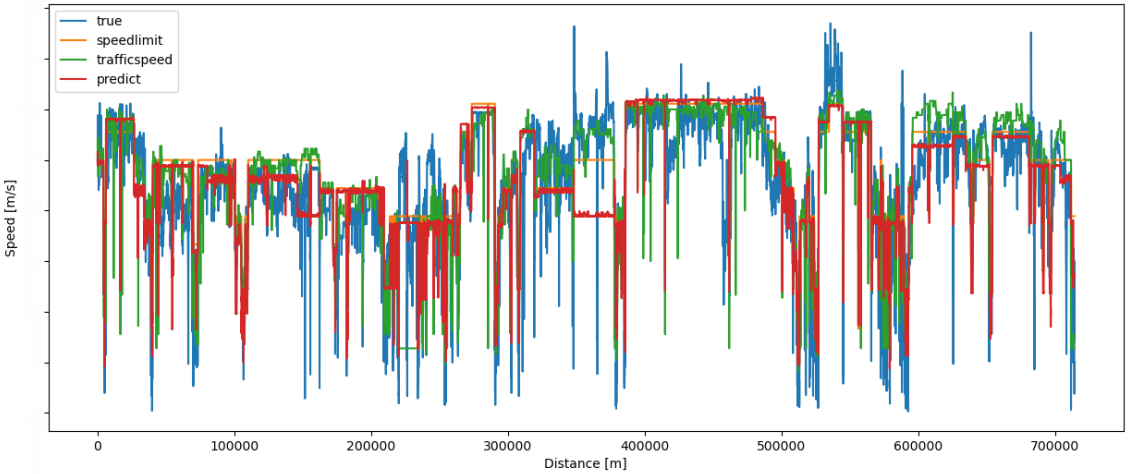


Figure 4.2: Speed prediction model many to one results compare with traffic speed from map

4.1.2 Speed Prediction Model (Many to many)

Based on the results in Table 4.2, we conclude that the LSTM model is the most accurate and reliable for speed prediction, as it has the lowest MSE (14.5142) and RMSE (3.8097), indicating superior overall predictive performance. Although the traffic speed model shows the lowest MAE (2.6384), suggesting minimal average deviation per prediction, it is slightly less effective than the LSTM model in handling overall error and outliers.

Table 4.2: Performance of speed prediction (Many to Many)

Model	MSE	MAE	RMSE
Speed limit	15.8263	2.8610	3.9782
Traffic speed	15.4275	2.6384	3.9278
LSTM	14.5142	2.6881	3.8097

Fig.4.3 indicates that the prediction model closely follows the actual speed trend for most of the distance, demonstrating good overall performance. However, there are noticeable deviations in certain segments where the predicted values significantly diverge from the actual values, suggesting that the model may struggle with high variability or extreme cases.

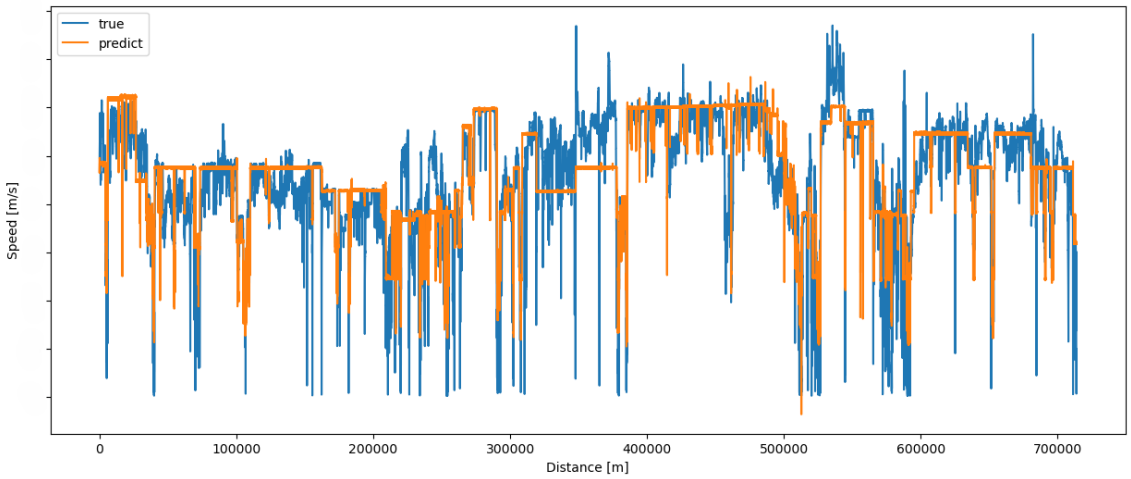


Figure 4.3: Speed prediction model many to many results

Comparing with API's data, Fig 4.4 shows that the predicted speeds (red) closely align with the true speeds (blue) across most distances, indicating good model performance. Both the traffic speed (green) and speed limit (orange) models generally follow the true speed trend but exhibit larger deviations in some areas. The predicted model outperforms the other two models by better capturing the variations in true speed, suggesting that it offers a more accurate and reliable prediction for different segments of the distance.

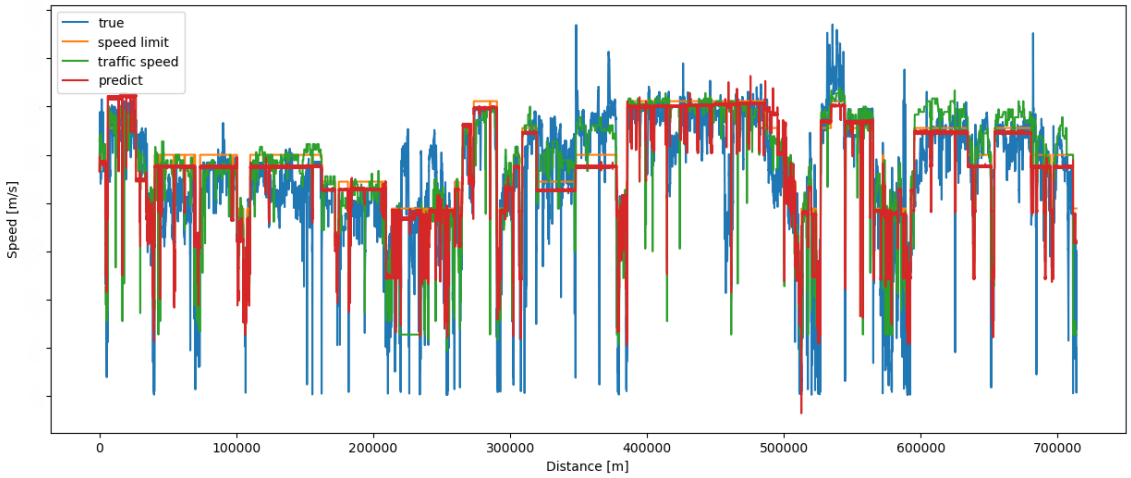


Figure 4.4: Speed prediction model many to many results compare with traffic speed from here map

4.1.3 Results Analysis

In the following sections, we will compare the performance of two different structural LSTM models and the ability of model generalization.

4.1.3.1 Error Analysis

Based on Table 4.3, the LSTM (many to many) model demonstrates superior performance in speed prediction compared to the LSTM (many to one) model. This is evident from its lower MSE of 14.5142, compared to 14.9930 for the many-to-one model, indicating better overall accuracy. Additionally, the many-to-many model has a lower MAE of 2.6881 compared to 2.7663, suggesting it has a smaller average deviation from the true values. Furthermore, RMSE for the many-to-many model is 3.8097, lower than the 3.8721 of the many-to-one model, indicating more precise and reliable predictions. Overall, the many-to-many architecture provides a more accurate and robust prediction framework for speed forecasting.

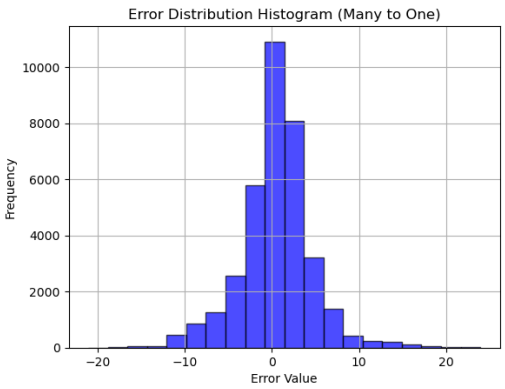
Table 4.3: Performance of different architectures for speed prediction

Model	MSE	MAE	RMSE
LSTM (many to one)	14.9930	2.7663	3.8721
LSTM (many to many)	14.5142	2.6881	3.8097

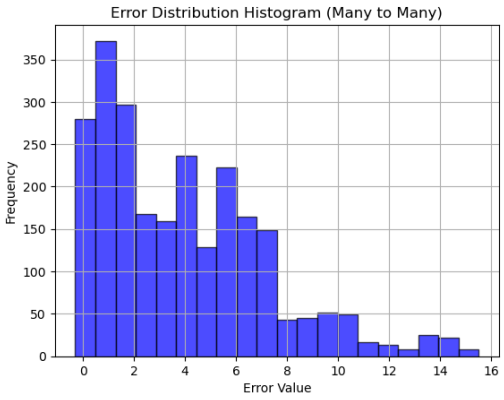
From Fig.4.5, we can conclude that the LSTM (many to one) model's error distribution is highly concentrated around zero, with a sharp peak at the center. This indicates that the model frequently makes small errors, with most predictions being very close to the actual values. The narrow spread around the center shows that the errors are generally consistent and do not vary widely. Besides, the LSTM (many to many) model has a wider and more dispersed error distribution. The errors are more spread out, indicating that this model makes a broader range of errors. While it does

show a peak at lower error values, it also has a significant number of higher error values, suggesting more variability and occasional larger errors in its predictions.

The many to one model has fewer instances of large errors, as evidenced by the steep decline in frequency as error values move away from zero. This model appears to be more stable and less prone to making large errors. In contrast, the many to many model shows a more gradual decline in frequency as the error values increase. There are more instances of larger errors compared to the many to one model, indicating that while it might capture more complex patterns, it also risks higher deviations. Overall, the many to many structure is better.



Many to one error distribution



Many to many error distribution

Figure 4.5: Histogram of error distribution

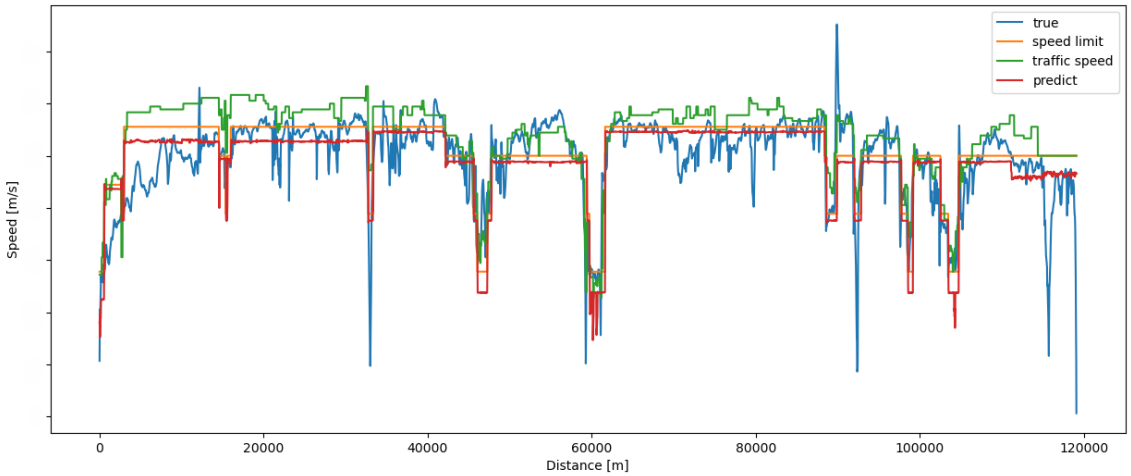
4.1.3.2 Analysis for the KDE covered by the training set

In KDE testing, most of the cases of the test set are included in the training set, and we analyse the prediction results of the two model structures.

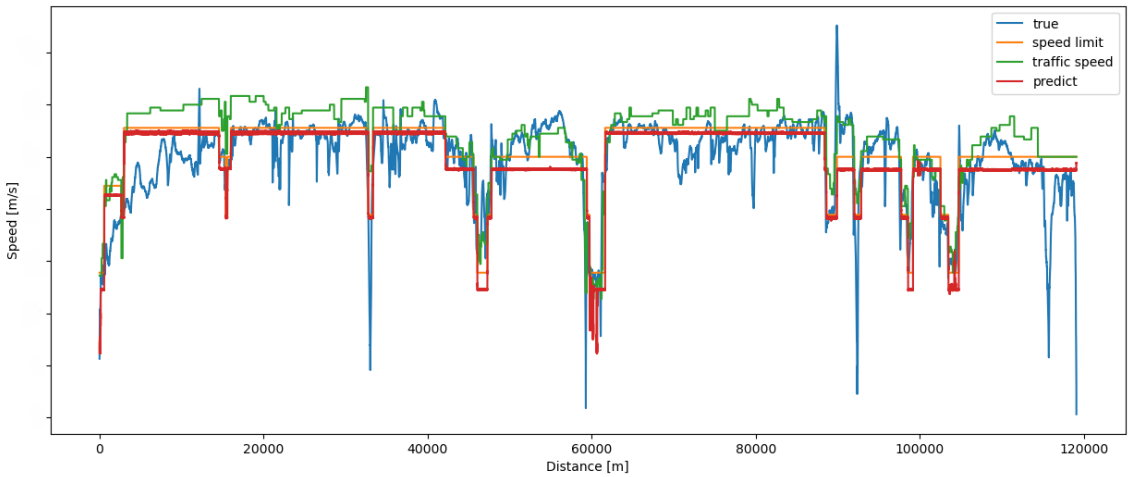
From the error matrix, it can be seen that all the error values of the many to many model are better than those of the many to one model, as shown in Table 4.4. Fig. 4.6 shows the prediction results of the two models in the same trip, and it can be seen that the model is able to achieve better prediction of the fluctuation of the speed in the structure of many to many struture. Fig.4.7 shows the frequency of the error distribution of the two models, and it can be obtained that the values of error around 0 are more frequent in the many to many structure. It can be concluded that the many to many structure has a higher frequency of values around 0. Therefore, the many to many model achieves better prediction results in the case of a better distribution of the test set.

Table 4.4: Performance of different architectures for speed prediction under cover

Model	MSE	MAE	RMSE
LSTM (many to one)	10.6543	2.3111	3.2641
LSTM (many to many)	9.0328	2.0620	3.0054

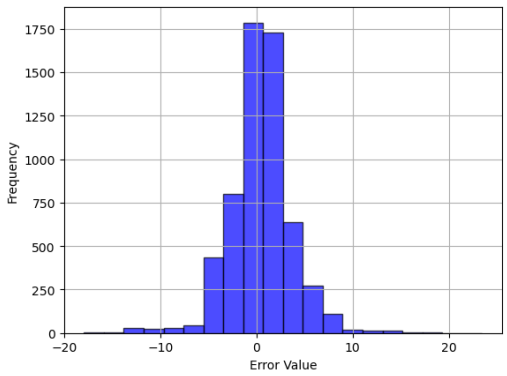


(a) Many to one model(trip 16)

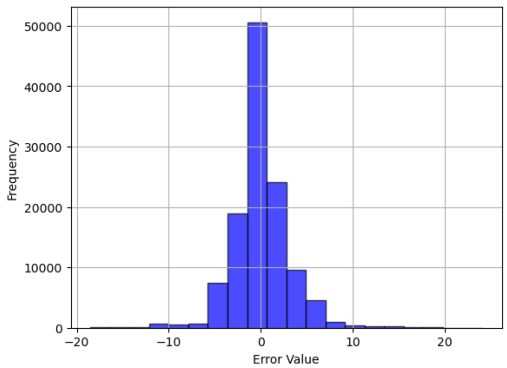


(b) Many to many model(trip 16)

Figure 4.6: Comparison of different model predictions under cover results



Many to one error distribution



Many to many error distribution

Figure 4.7: Histogram of under cover error distribution

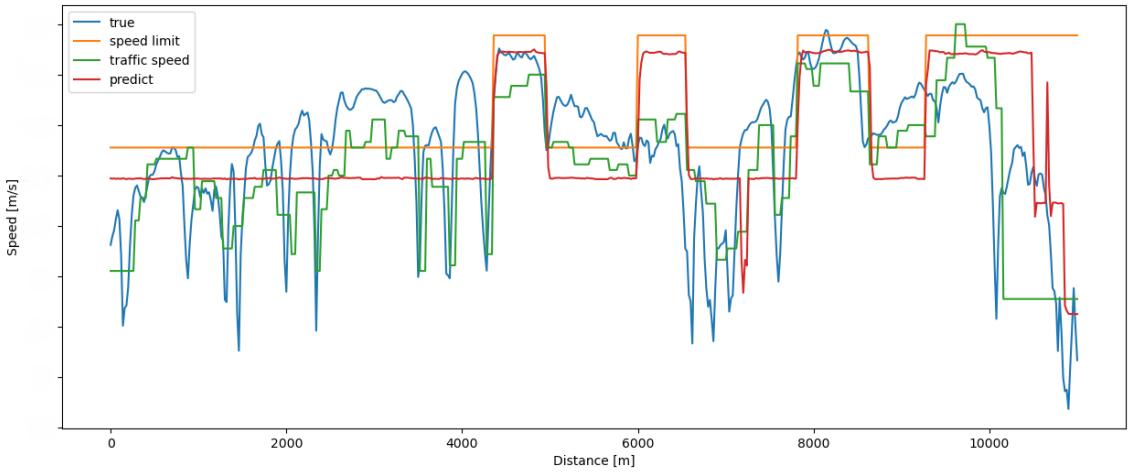
4.1.3.3 Analysis for the KDE not covered by the training set

During the KDE analysis, we found that some trips kept a constant value for the time after the encode and thus could not be viewed in the KDE distribution graphs. This also means that these times are a constant time, whereas the time distribution our training set are varied. Therefore, it is necessary to check results for such test data which have large differences from the training set distribution. If the distribution of the test data is significantly different from that of the training set, but the model is able to generate better predictions, this indicates that the model is more generalisable.

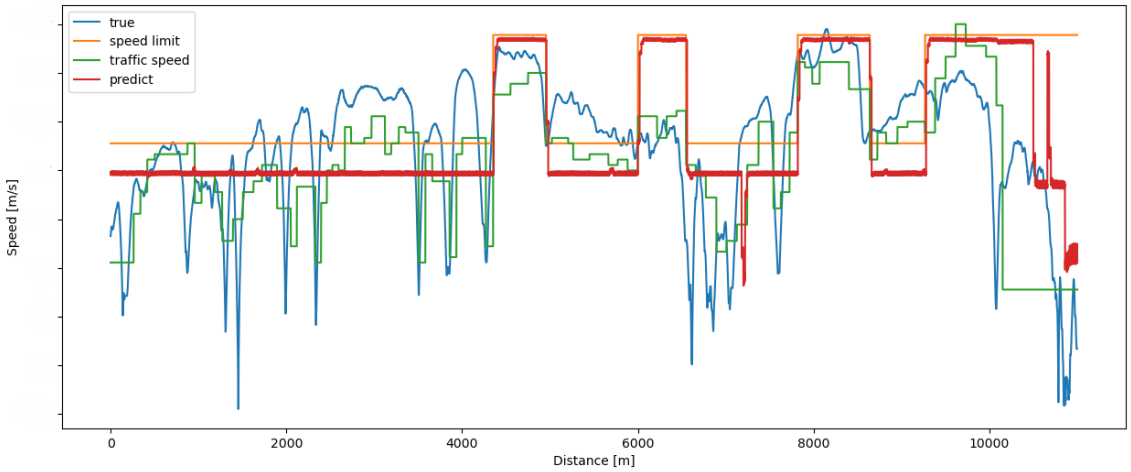
From Table 4.5, it can be concluded that the value of the error matrix becomes larger compare with Table 4.4, which means that the model's predictive ability becomes worse when predicting these poorly distributed trips. Comparing the error matrices of the two models, the many to many structure slightly better than many to one. from Fig.4.8, we can see that the prediction ability of the two models is similar, but the many to many structure can still capture some speed fluctuations. In this situation, the model's prediction is almost entirely dependent on the speed limit. In the error distribution plot, as shown in Fig. 4.9, the error range distribution of these two models is significantly expanded, but the many to many structure still has a lower prediction error.

Table 4.5: Performance of different architectures for speed prediction not under cover

Model	MSE	MAE	RMSE
LSTM (many to one)	13.5673	3.0820	3.6833
LSTM (many to many)	13.1895	2.9704	3.6317

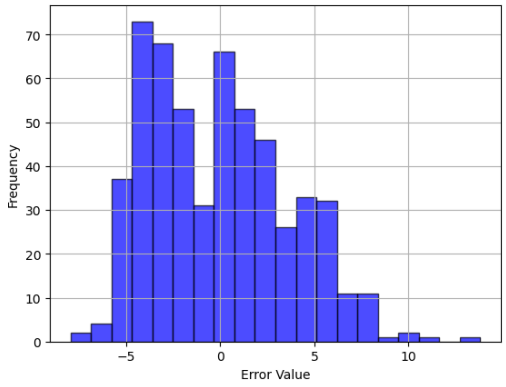


(a) Many to one model(trip 3)

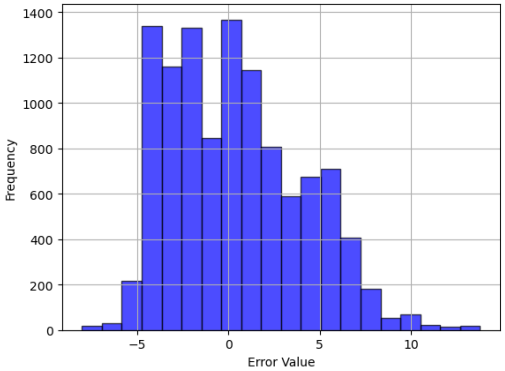


(b) Many to many model(trip 3)

Figure 4.8: Comparison of different model predictions not under cover results



Many to one error distribution



Many to many error distribution

Figure 4.9: Histogram of not under cover error distribution

In conclusion, the many to many structure has better prediction results, especially

when the test set is well distributed, and this model structure can better capture the trend of speed change and generate lower errors.

4.2 Energy Consumption Prediction Model

For energy consumption prediction, we have divided the prediction method into instant energy consumption and total energy consumption. We use the energy consumption of the resistance as our baseline.

4.2.1 Predict Instant Energy Consumption

4.2.1.1 Feature Selection

There are different ways of selecting features. We chose as input the features which have higher correlation in the correlation matrix, as shown in Table 4.6 In addition,

Table 4.6: Instant Energy Consumption Prediction Model Features Selection (8 Inputs)

Input	Speed
	Temperature at 2m
	Wind speed at 10m
	Wind direction at 10m
	Traffic speed
	Speed limit
	Category
	Instant resistance energy consumption
Output	Instant energy consumption

we added some parameters that are correlated but not highly relational as features for training the model, as shown in Fig.4.7.

Through the VIF test, we removed the parameters with strong multicollinearity and kept only five features as inputs,as shown in Fig.4.8. In the following sections, we will compare and discuss the impact of the different feature inputs on the model results.

4.2.1.2 Model evaluation

In this section, we compare the prediction performance of the base machine learning model and the neural network model for different input situations respectively, by using the dynamics model of the vehicle as the baseline.

When choosing 8 features as inputs, as shown in Table 4.9 all machine learning models except the neural network model outperform baseline, so we do not consider the NN model in the following comparisons.

Table 4.7: Instant Energy Consumption Prediction Model Features Selection (13 Inputs)

Input	Speed
	Temperature at 2m
	Wind speed at 10m
	Wind direction at 10m
	Traffic speed
	Speed limit
	Category
	Type
	Curve
	Week
	Day
	Minute
	Instant resistance energy consumption
Output	Instant energy consumption

Table 4.8: Instant Energy Consumption Prediction Model Features Selection (5 Inputs)

Input	Speed
	Temperature at 2m
	Vehicle Acceleration
	Slope
	Instant resistance energy consumption
Output	Instant energy consumption

Table 4.9: Performance of different regression algorithms for the instant energy consumption (8 input)

Model	MSE	MAE	RMSE
RandomForestRegression	1.1287e-7	0.00010429	0.0003359
LinearRegression	1.517e-7	0.0001399	0.0003894
DecisionTreeRegressor	2.0919e-7	0.0001349	0.0003462
KNeighborsRegressor	1.198e-7	0.0001089	0.0003462
Ensemble Learning	1.233e-7	0.0001039	0.0003512
NN	1.233e-7	9.097e-07	0.0009537
Physical Model(baseline)	6.5692e-7	0.0002001	0.0008105

Most of the machine learning models improved their accuracy when we chose 5 features as inputs,as shown in Table 4.10 but the decision tree regression model had a slightly raised error value.

Table 4.10: Performance of different regression algorithms for the instant energy consumption (5 input)

Model	MSE	MAE	RMSE
RandomForestRegression	9.0721e-8	0.0001113	0.0003012
LinearRegression	1.0731e-7	0.0001434	0.0003273
DecisionTreeRegressor	2.0957e-7	0.0001456	0.0004579
KNeighborsRegressor	1.190e-7	0.0001215	0.0003450
Physical Model(baseline)	6.5692e-7	0.0002001	0.0008105

When we choose 13 features as inputs, as shown in Table 4.11 the errors of almost all models are increased, especially the K neighbours regressor. However, the accuracy of the decision tree regression is a little increased compared to select 5 features as inputs.

Table 4.11: Performance of different regression algorithms for the instant energy consumption (13 input)

Model	MSE	MAE	RMSE
RandomForestRegression	1.4447e-7	0.00010961	0.0003801
LinearRegression	6.8678e-7	0.0001451	0.0002621
DecisionTreeRegressor	1.5564e-7	0.0001395	0.0003945
KNeighborsRegressor	2.0817e-7	0.0001148	0.0004563
Physical Model(baseline)	6.5692e-7	0.0002001	0.0008105

As can be learnt from the above results of instant energy consumption predictions, the prediction errors of all the models except the NN model are relatively small. Then, we add up each instant energy consumption to get the value of the total energy consumption and compare the prediction performance of these models.

When choosing 5 features as inputs, as shown in Table 4.12 we found that most of the models predicted better than baseline. However, the ensemble learning and NN models did not predict as well as baseline, where the RMSE of the NN model was greater than that of baseline by nearly 1 kWh. Therefore, we will not consider NN model and ensemble learning in the following prediction method.

Table 4.12: Performance of different regression algorithms for the add all instant energy consumption (8 input)

Model	MSE	MAE	RMSE
RandomForestRegression	2.8669	1.1706	1.6931
LinearRegression	2.3269	1.1878	1.5254
DecisionTreeRegressor	2.4030	1.0229	1.5502
KNeighborsRegressor	1.9795	1.0474	1.4069
Ensemble Learning	4.6147	1.4478	2.1482
NN	10.2185	2.3611	3.1966
Physical Model(baseline)	4.4781	1.3564	2.1161

When we reduced the number of input features and selected only 5 features as inputs, as shown in Table 4.13, the performance of the model got a significant improvement. Especially for the decision tree regressor, the RMSE is only 0.69kWh.

Table 4.13: Performance of different regression algorithms for the add all instant energy consumption (5 input)

Model	MSE	MAE	RMSE
RandomForestRegression	1.0943	0.7485	1.0461
LinearRegression	1.145	0.8031	1.0700
DecisionTreeRegressor	0.4795	0.4838	0.6925
KNeighborsRegressor	0.5557	0.5157	0.7454
Physical Model(baseline)	4.4781	1.3564	2.1161

When we increase the number of features to 13, as shown in Table 4.14, the prediction performance of the model decreases significantly, especially for the decision tree regressor, where the model's prediction error goes beyond the baseline.

Table 4.14: Performance of different regression algorithms for the add all instant energy consumption (13 input)

Model	MSE	MAE	RMSE
RandomForestRegression	2.7229	1.2401	1.6501
LinearRegression	1.4172	3.6373	1.9072
DecisionTreeRegressor	8.2014	1.6563	2.8638
KNeighborsRegressor	3.7886	1.2937	1.9464
Physical Model(baseline)	4.4781	1.3564	2.1161

From the above analysis we can get that the decision tree regressor has the best prediction performance when the input features are 5.

4.2.1.3 Model Prediction Visualisation

The two histograms in the figures (as shown in 4.12) represent the error distribution for a baseline model and a Decision Tree Regressor model. From these plots, it is evident that the Decision Tree Regressor model has a more concentrated error distribution around zero, indicating a lower variance and more consistent predictions compared to the baseline model. The baseline model shows a wider spread of error values, suggesting it has less predictive accuracy. Therefore, the Decision Tree Regressor model demonstrates a better performance in terms of error reduction compared to the baseline model.

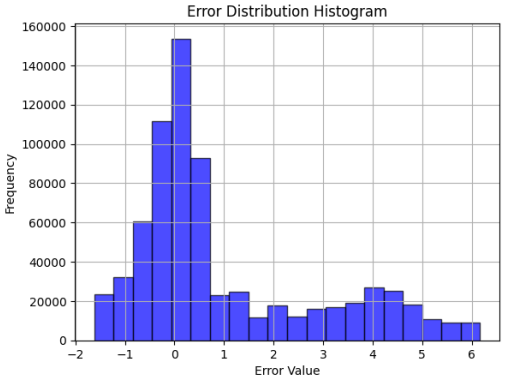


Figure 4.10: Baseline error distribution

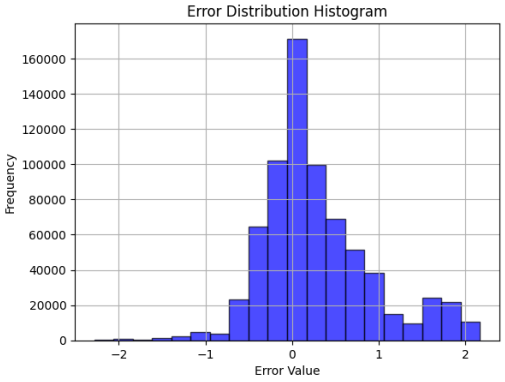


Figure 4.11: Decision Tree Regressor model error distribution

Figure 4.12: Error distribution comparison of baseline and best prediction model

From the visualized prediction figures, it can be concluded that the model has a better prediction ability when the model predicts trips with shorter distances, as shown in Fig.4.13. When the model predicts trips of longer distances, the predictive capacity of the model becomes poor. For example, when the trip is about 40 kilometers, the model's predictions are not desirable and have a large deviation,as shown in Fig.4.14.

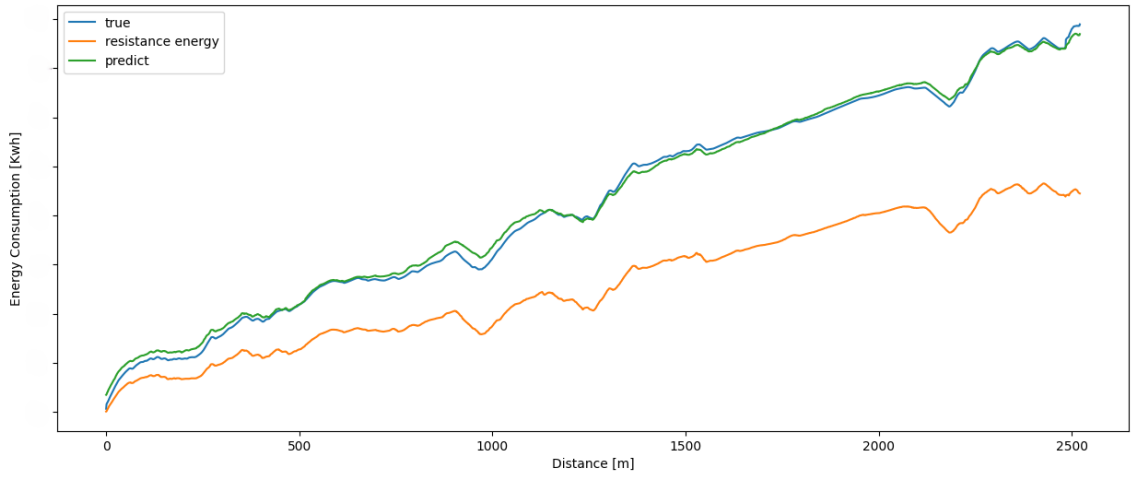


Figure 4.13: Short trip 17

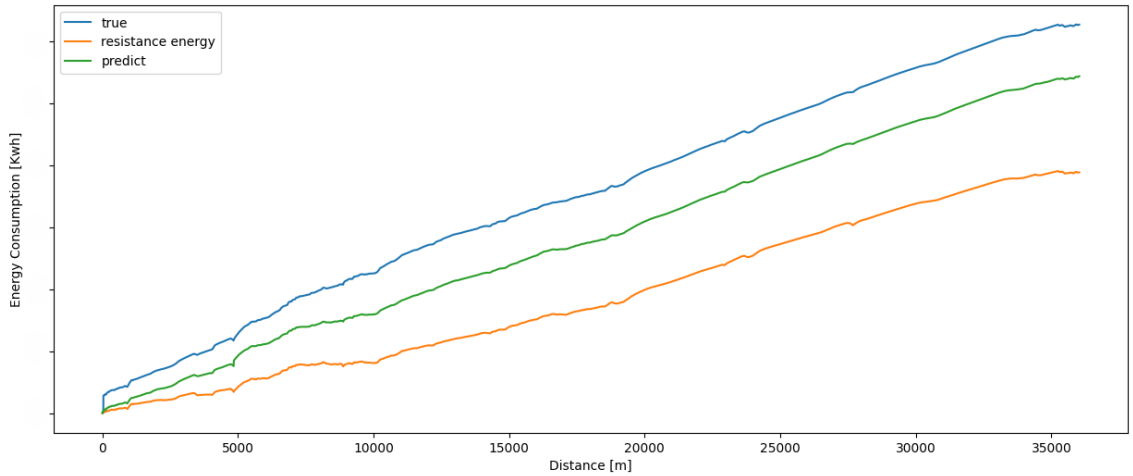


Figure 4.14: Long trip 6

Generally, when processing longer time series prediction problems, deep learning methods such as LSTM are used to solve such problems. In the following sections, we will discuss the performance of LSTM models in making total energy consumption predictions.

4.2.2 Predict Total Energy Consumption

In our prediction of total energy consumption of vehicles, we have tried to use base machine learning models for prediction and most of the base models have lower prediction performance than deep learning models,as shown in Table 4.15. Although the prediction error of Linear Regression model is lower than LSTM model, while training the base model, our dataset is downsampled processed dataset with scientific distribution based on Kmeans clustering. Therefore, while these data contain almost all the situations in the original training set, the total amount of data is much lower than the original amount of data, and these data lack more detailed information. However, for deep learning models, we can use all of the data as a training set by using a sliding window to take values. So, in the following sections we use the CNN-LSTM method to predict.

For total energy prediction, we used two structures of LSTM for prediction, that are many to one and many to many. In the following subsections, we will analyze the advantages and disadvantages of the prediction results of the two model structures with different inputs. The different feature inputs and the reasons for their selection will be explained more in many to many section.

Table 4.15: Performance of different regression algorithms for the total energy consumption (8 inputs).

Model	MSE	MAE	RMSE
LinearRegression	2.4552	1.2995	1.5669
DecisionTreeRegressor	3.7924	1.3749	1.9474
KNeighborsRegressor	26.1091	3.6818	5.1097
Ensemble Learning	1716.3733	39.219	41.429
CNN + LSTM(many to one)	3.9267	1.4421	1.9816
CNN + LSTM(many to many)	3.0109	1.2133	1.7352

4.2.2.1 Many to one

In the many to one model, the error matrix shows that the prediction values of the models trained with the three different feature inputs are better than those calculated by the basic physical model, as shown in Table 4.16. The lowest prediction error value of the model is obtained when the input feature is 13.

Table 4.16: Performance of different inputs for total energy prediction (Many to one)

Model	MSE	MAE	RMSE
CNN + LSTM (many to one, 8 inputs)	3.9267	1.4421	1.9816
CNN + LSTM (many to one, 13 inputs)	2.0339	1.0827	1.4261
CNN + LSTM (many to one, 5 inputs)	3.7779	1.4136	1.9437
Physical Model(baseline)	4.4781	1.3564	2.1161

The Fig.4.15 shows error distribution histograms for energy prediction models with different input features (5, 8, and 13 inputs) in a many to one model. As the number of inputs increases from 5 to 13, the spread of the errors becomes more concentrated around zero, particularly noticeable in the histogram for 13 inputs, which shows a more pronounced peak and narrower distribution. This suggests that the prediction accuracy improves with more inputs, as the models with higher input complexity produce fewer large errors and exhibit a tighter error distribution.

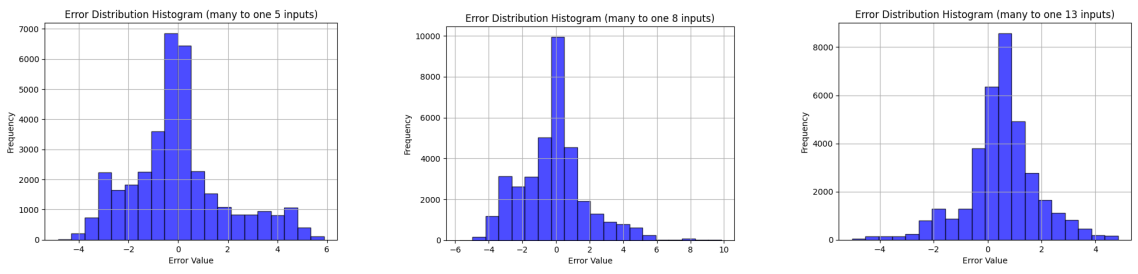


Figure 4.15: Error Distribution Histogram for Energy Prediction (many to one)

4.2.2.2 8 inputs

Compared to the many to many model, many to one does not predict as well as it does, as shown in Fig.4.16. The predicted values are almost similar to the baseline and have a large gap between the true values.

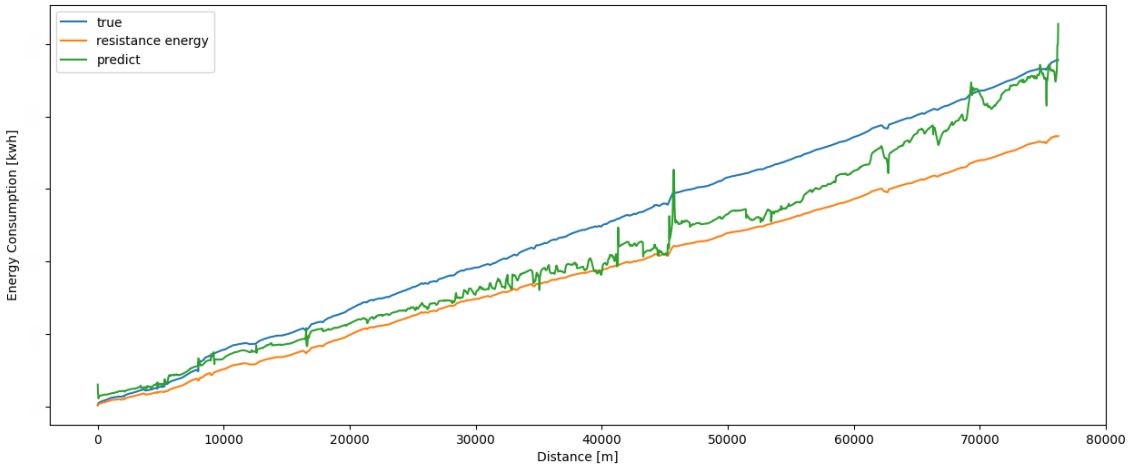


Figure 4.16: 8 Inputs Predict Results (trip 8)

4.2.2.3 13 inputs

When using the many to one model, better predictions can be achieved by using 13 features as inputs. Compared to many to many model, many to one model (in Fig.4.17) is able to predict better and accurate energy consumption.

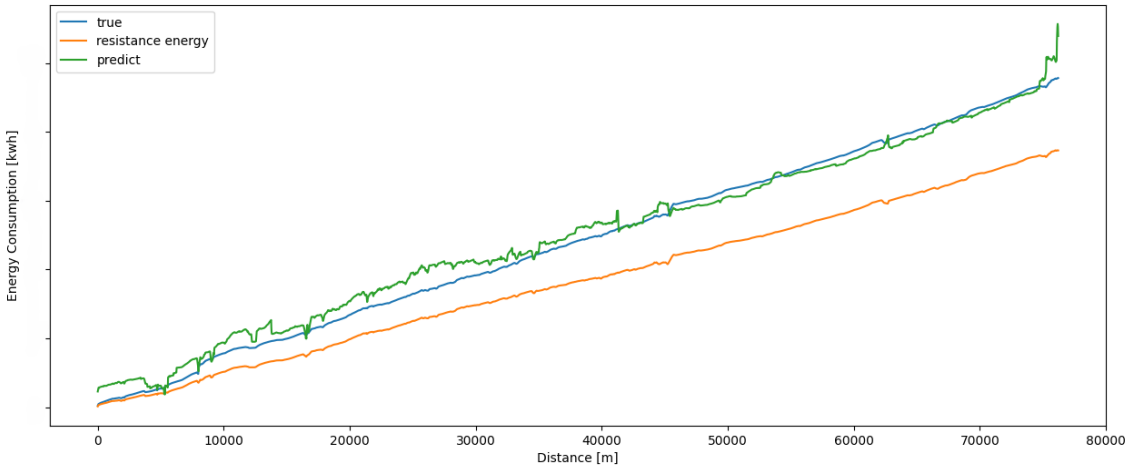


Figure 4.17: 13 Inputs Predict Results (trip 8)

4.2.2.4 5 inputs

When using 5 features as inputs, the model's predictions showed large fluctuations, as shown in Fig.4.18.

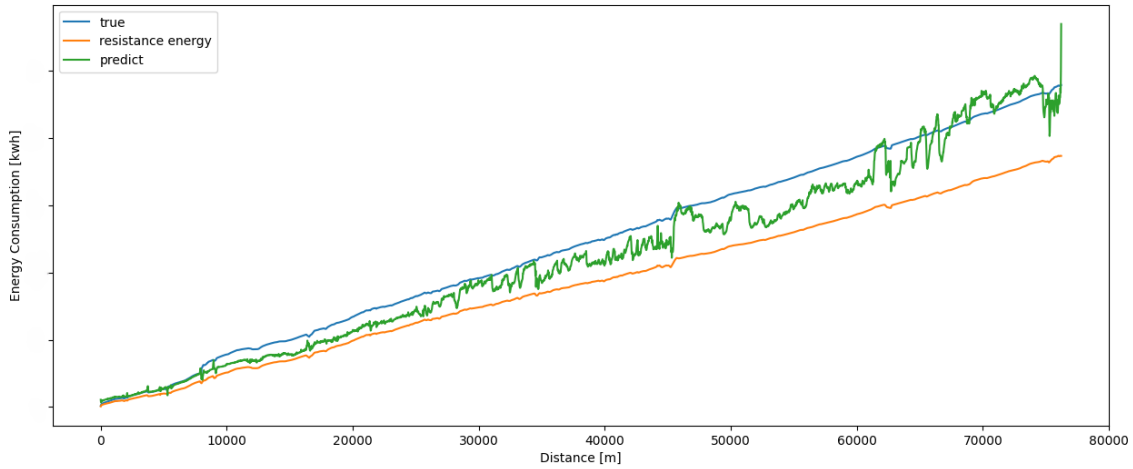


Figure 4.18: 5 Inputs Predict Results (trip 8)

It can be found from the above error matrix and the actual energy consumption prediction plot that when using the many to one structure, better prediction results can be achieved by using 13 features as inputs.

4.2.2.5 Many to many

In the many to many structure, we have tried to train the model using different inputs and the error matrix is shown in Table 4.17. From Table 4.17 we can conclude that CNN + LSTM (many to many) model with 5 inputs performs the best for total energy prediction, as it has the lowest MSE (1.9095), MAE (1.0442), and RMSE (1.3818). In contrast, increasing the number of inputs to 13 results in the worst performance, with the highest errors across all metrics. The model with 8 inputs performs moderately, with better accuracy than the 13-input model but not as good as the 5-input model. This suggests that a more complex model with more inputs does not necessarily lead to better performance, and in this case, a simpler model with fewer inputs provides more accurate and reliable predictions.

Our calculated error is the average of the errors of all trips. In the following section, we will analyse the prediction results of the models trained by different inputs and analyse the results.

Table 4.17: Performance of different inputs for total energy prediction (Many to many)

Model	MSE	MAE	RMSE
CNN + LSTM (many to many, 8 inputs)	3.0109	1.2133	1.7352
CNN + LSTM (many to many, 13 inputs)	3.9978	1.2699	1.9994
CNN + LSTM (many to many, 5 inputs)	1.9095	1.0442	1.3818
Physical Model(baseline)	4.4781	1.3564	2.1161

4.2.2.6 8 inputs

First we chose as input the features that have higher correlation in the correlation matrix, as shown in Table 4.18

Table 4.18: Total Energy Consumption Prediction Model Features Selection (8 Inputs)

Input	Speed
	Temperature 2m
	Wind speed 10m
	Wind direction 10m
	Traffic speed
	Speed limit
	Category
	Total resistance energy consumption
Output	Total energy consumption

From Fig.4.19, we can conclude that the predicted energy consumption (green) closely follows the actual energy consumption (blue) across the entire distance, but tends to underestimate energy consumption slightly compared to the true values.

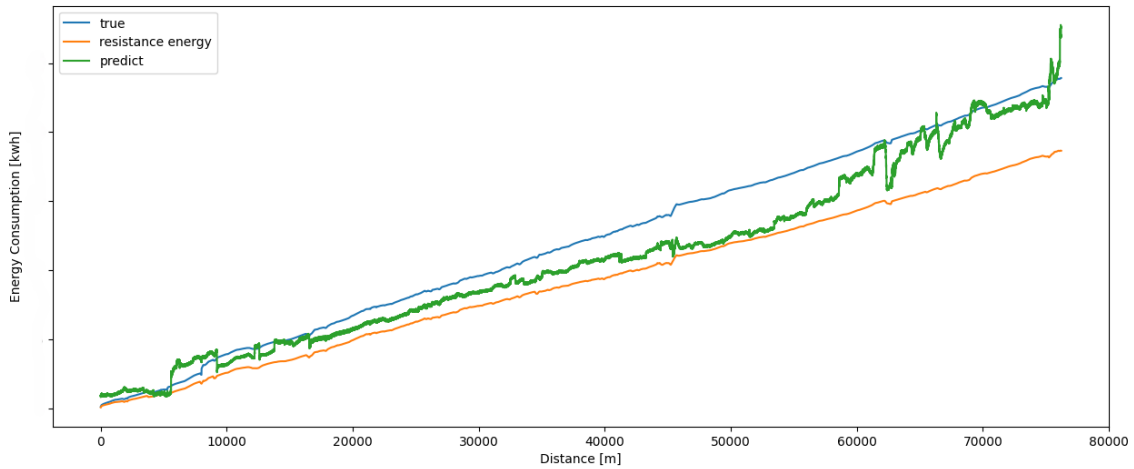


Figure 4.19: 8 Inputs Predict Results (trip 8)

4.2.2.7 13 inputs

In order to make the model predictions more accurate, we added additional parameters related to the road. From Fig.4.20, we can conclude that the predicted energy consumption (green) is closer to the actual energy consumption (blue) throughout the distance than the resistance energy (red). In the selected trip, there is a large gap between the resistance's energy consumption and the real energy consumption, which is well compensated for by our model prediction.

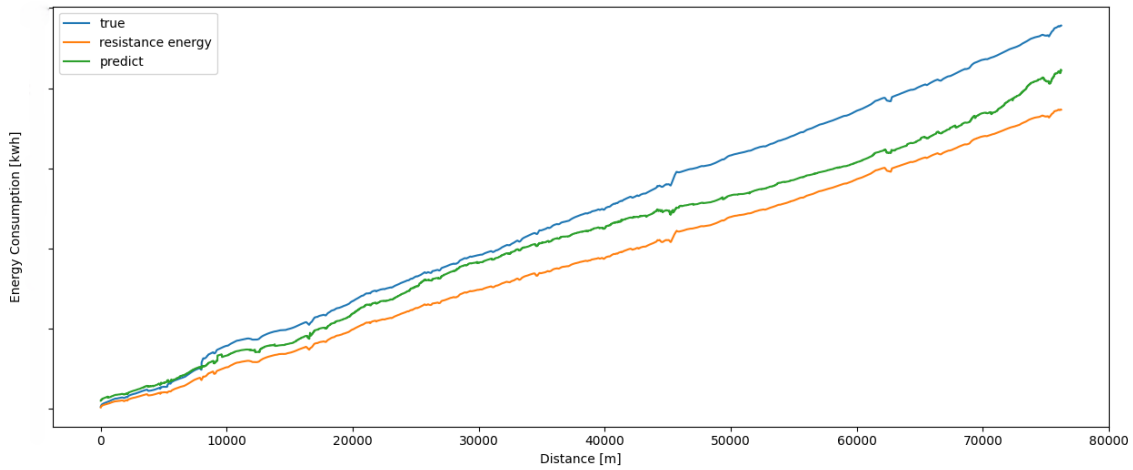


Figure 4.20: 13 Inputs Predict Results (trip 8)

4.2.2.8 5 inputs

Through VIF testing, we use the five features shown in Table 4.19 as inputs.

Table 4.19: Total Energy Consumption Prediction Model Features Selection (5 Inputs)

Input	Speed
	Temperature 2m
	Vehicle Acceleration
	Slope
	Total resistance energy consumption
Output	Total energy consumption

Fig.4.21 shows the predicted values for the selected trips and it can be found that this model also compensates for the difference between resistance's energy consumption and the total energy consumption.

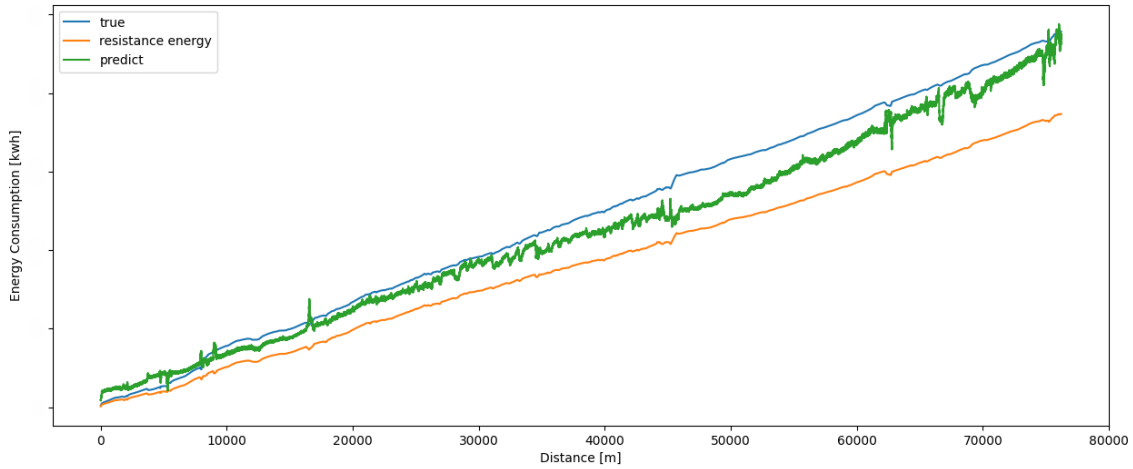


Figure 4.21: 5 Inputs Predict Results (trip 8)

4.2.2.9 Results Analysis

The Fig.4.22 shows error distribution histograms for energy prediction models with different input features (5, 8, and 13 input features) in a many-to-many model architecture. The central peaks of the histograms are all around zero, indicating that the models generally predict values close to the actual values. However, as the number of inputs increases from 5 to 13, the spread of errors appears to increase slightly, suggesting that models with more inputs may have a wider range of errors but still maintain a central tendency near zero. This implies that while the models are overall accurate, increased input complexity may introduce slightly more variability in the predictions. The model predicts best when the input features are 5.

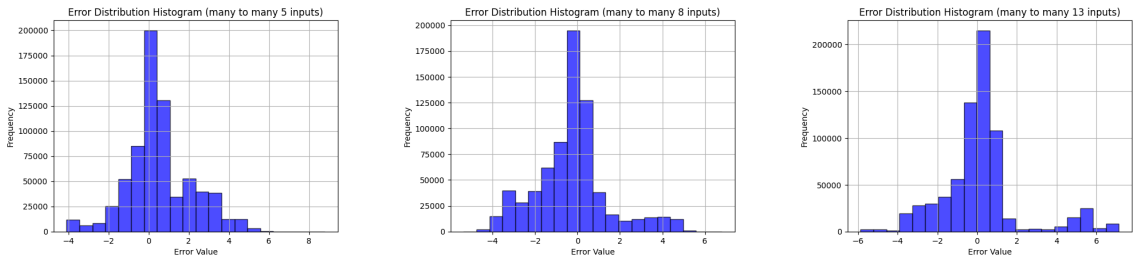
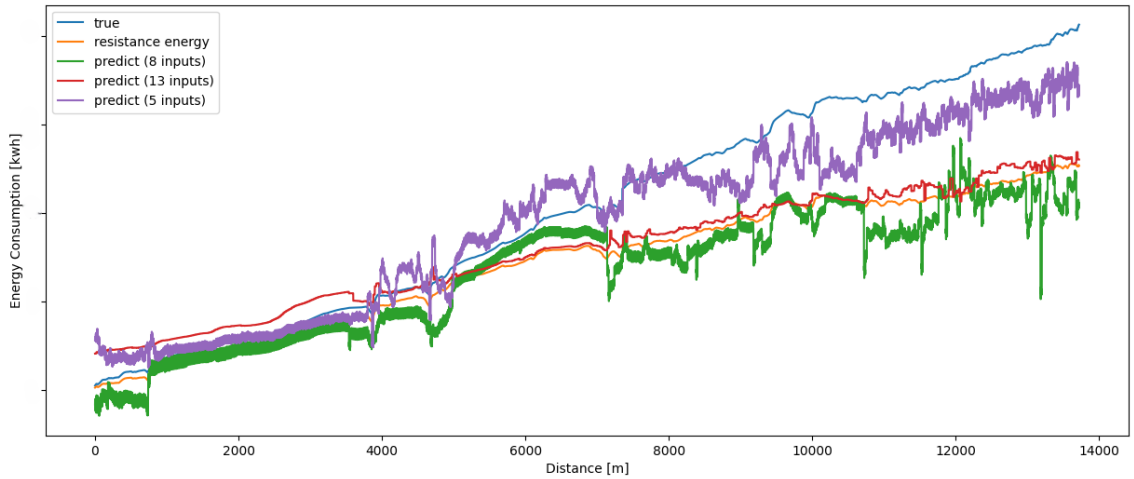
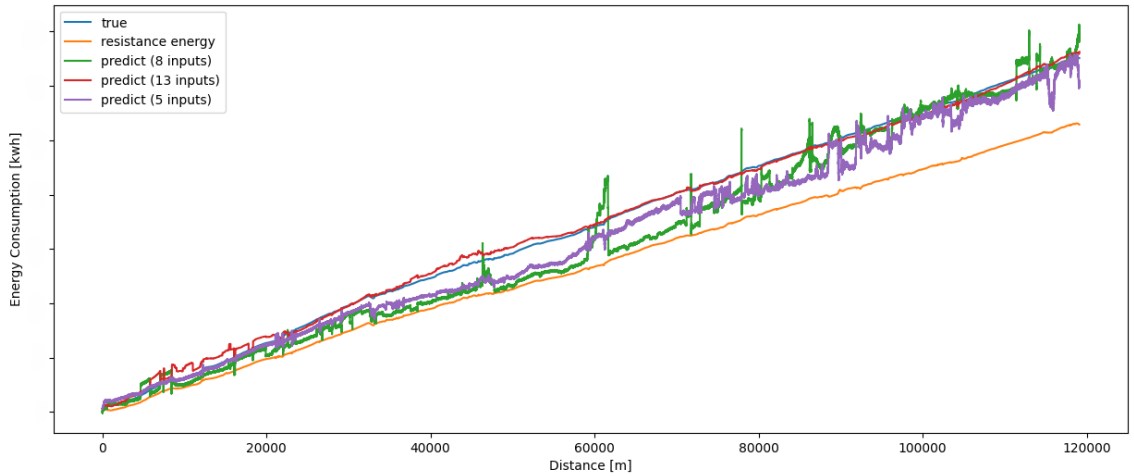


Figure 4.22: Error Distribution Histogram for Energy Prediction (many to many)



(a) Short trip 11



(b) Long trip 16

Figure 4.23: Comparison of different model predictions for different lengths of trips.

We summarize these three models and analyze them comparatively according to the length of the trips. The 13 inputs model (red) typically provides the most accurate and consistent predictions for long-distance trips, closely correlating with real energy consumption. The model with 5 inputs (purple) also performs well and usually provides the most accurate predictions for short trips, as can be seen in the figure with three different inputs, only in the 5-input case the predictions successfully compensate for the gaps in resistance energy consumption and total energy consumption, but with some fluctuations and occasional overestimation. The model with 8 inputs (green) shows greater fluctuations and occasional overestimation.

4.2.3 Best Prediction Model Comparison

Comparing the models which have the best predictions in each format, we found that when predicting long distance trips, the best predictions were made by using the CNN-LSTM model(5 inputs with many to many structure achieves best results), as

shown in Fig.4.24. When predicting trips over short distances, the best predictions were achieved by using Decision tree regression (5 inputs features and predict instant energy), as shown in Fig.4.25. There is a special situation where both the models give better predictions when the vehicle's total energy consumption and resistance energy consumption are similar, as shown in Fig.4.26.

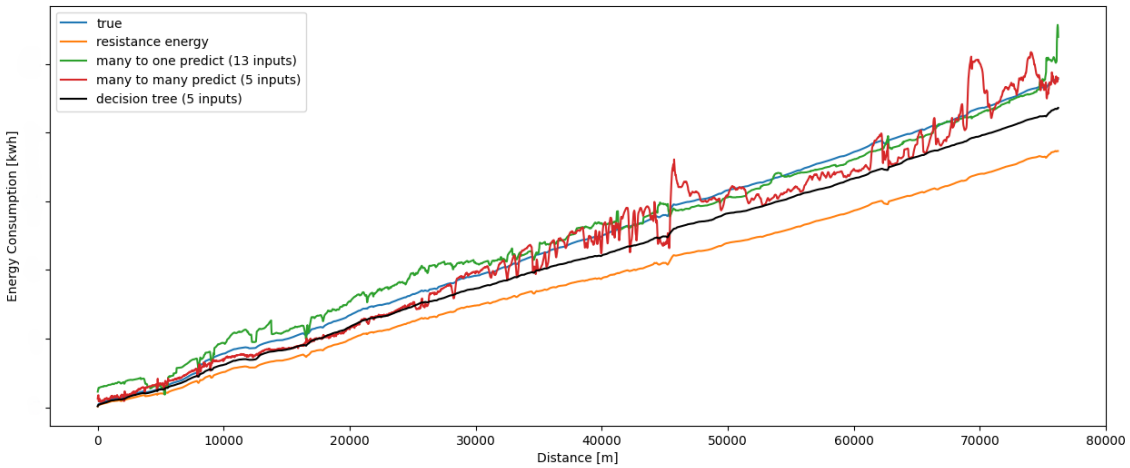


Figure 4.24: Long Trip Predict Results (trip 8)

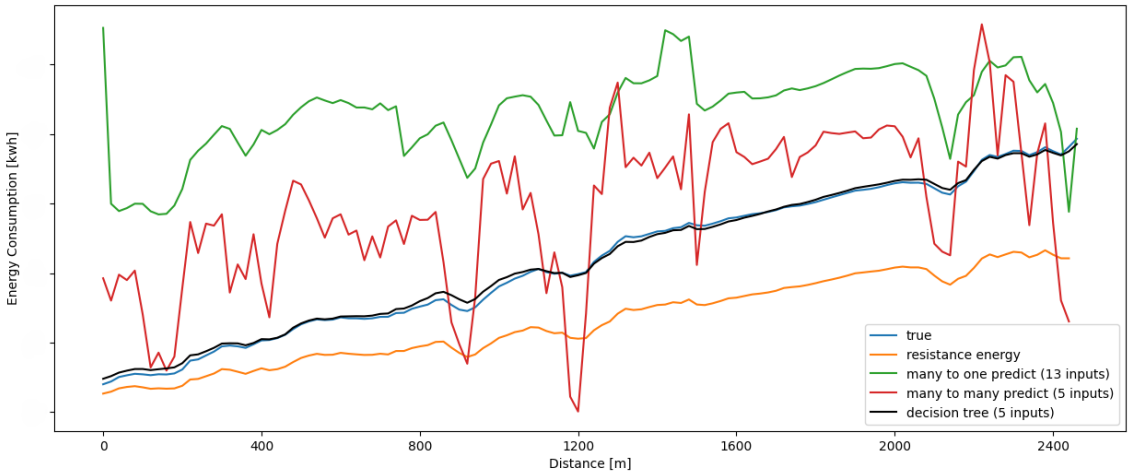


Figure 4.25: Short Trip Predict Results (trip 17)

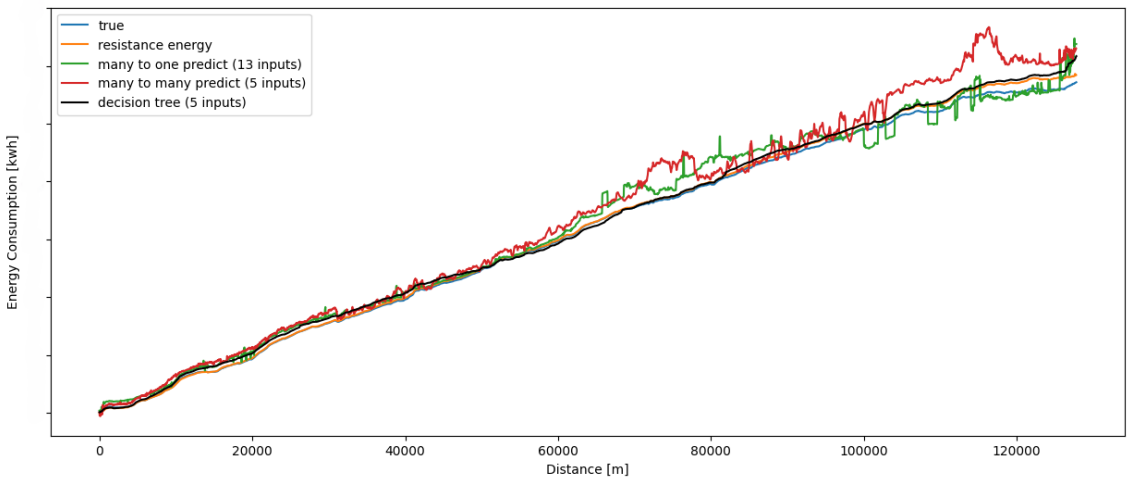


Figure 4.26: Similar Energy Consumption and Resistance Energy consumption Predict Results (trip 10)

5

Conclusion

5.1 Discussion

5.1.1 Vehicle Dynamic Model

The vehicle dynamics model is mainly used to calculate the energy consumption due to travelling resistance. In our model we take into account more detailed considerations, for example in the calculation of the air resistance we take into account the calculation of the relative speed. By comparing the resistance energy consumption with the total energy consumption, we find that the trend of the two energy consumptions is the same but the gap between the two energy consumptions gradually becomes larger as the travelling time becomes longer. This may be caused by the accumulation of errors.

5.1.2 Speed Prediction Model

In the speed prediction model, the many to many structure predicts the best results in the LSTM model and our predicted values beat the predicted values of traffic speed from map API. The traffic speed, which was previously the main speed input in energy consumption predictions, has been beaten by our model. This enhancement enables the company to achieve more accurate predictions and reduce reliance on external APIs. We also beat the speed prediction model built by Ziegmann et al. [19], which had an RMSE of 3.864, while our model had a better RMSE of 3.8097.

By plotting the predictions of LSTM with two different output structures, we can conclude that the predictions become fluctuate when using the many to many structure. In the speed prediction model, the speed limit is an important feature. Since the speed limit is a constant value on most roads, this leads to the predicted speed remaining constant for most of the time. The predicted value sometimes changes abruptly, it is possible that the vehicle is travelling around a corner and the speed limit suddenly decreases. At some moments, the vehicle's speed is greater than the speed limit. Such instances are very rare. Our model predicts driving that follows legal speed limits, and it is unlikely to predict instances where someone might choose to exceed speed limits. From the map traffic speed, it can be noticed that in some roads the traffic speed is also greater than the speed limit, which mostly happens in the early morning and at night when there are fewer vehicles on the road.

The distribution of the test set has a large impact on predictive capacity of the model. When constructing the training and test sets, it is important to make the data distribution of the training set as broad as possible.

5.1.3 Energy Consumption Prediction Model

In the energy consumption prediction model, the model's prediction performance depends on the trip's length and the number of the inputs. In shorter trips, the decision tree regression model with 5 inputs can achieve better prediction results in predicting instant energy consumption, and in longer trips, the many to many of CNN-LSTM with 5 inputs can make better prediction results. This may be because our LSTM model is better suited for long-range prediction, which leads to overfitting when predicting short range trips where it captures the noise in the data. In addition, in the short range trip, trips are more homogeneous in terms of environmental factors. Decision trees can effectively make decisions based on these features. We pre-prune the decision tree by setting the maximum depth to prevent it from overfitting. In the prediction of energy consumption for long-range trips, it is necessary to consider the effects of longer historical data and future states. For example, predicting the energy consumption of a long trip may need to consider the combined effects of a number of factors, such as the starting state, environmental changes, duration, etc., and LSTM is better able to capture the temporal relationships between these factors.

The fluctuation of the predicted values is almost zero compared to the LSTM model in the prediction using the classical machine learning model. In contrast, the prediction of the deep learning model has large fluctuations. This fluctuation is most likely due to the high sampling frequency of our data, which results in the sudden change in the presence of certain values and the model becoming jittery. In addition, the accuracy of the deep learning model is higher than the regular machine learning model. When training the model, we added the energy consumption caused by driving resistance as a feature input to the model, which also made our model predictions more accurate. However, most of our dataset is collected from November to May, when the temperature is lower, and it is temporarily impossible to verify the prediction effect of the model when the temperature is warmer.

5.2 Conclusion

Considering people's range anxiety during driving electric vehicles, this thesis proposes two LSTM-based prediction models for predicting vehicle speed and total energy consumption respectively. The dataset is acquired from Zeekr's vehicle and external APIs.

In the speed prediction model, compare the many to one and many to many structures constructed based on the LSTM model respectively. The many to many structure achieves the best prediction results, with an RMSE of 3.8097 for its prediction structure, and this prediction result is better than the prediction results of the map API and the results achieved in published literature [19].

In the energy prediction model, there are two methods proposed for energy prediction, which are predicting the instant energy consumption and summing up at the end to calculate the total energy consumption, and the other method is to predict the total energy consumption directly. Considering the complexity of the data, in addition to use the basic machine learning regression model, a CNN - LSTM method is proposed for the prediction the total energy consumption. Considering the impact of different feature inputs on the model, the input features are classified as 8, 13, and 5 based on correlation matrix and multicollinearity testing respectively.

Based on the prediction results of different models, it is shown that when the trip's journey is short, using 5 features as inputs, decision tree regression model for predicting instant energy consumption can achieve better prediction results with an RMSE of 0.6925.When the predicted trip's journey is long, the CNN-LSTM many to many structure achieves the best results in predicting the total energy consumption, where the RMSE is 1.3818 when 5 feature are used as input.

Since the accuracy of the speed prediction model is not very high, there are discrepancies in the whole prediction process. In future work, the prediction accuracy of the speed prediction model needs to be further improved, for example, it needs to take more account of driver behaviour (for instance, records acceleration patterns, braking habits, and steering maneuvers), road conditions (for instance, traffic signal), etc. Then, the predicted speed can be used as an input to the energy consumption model for energy consumption prediction. In addition, a suitable distance threshold has to be found in the energy prediction model to distinguish between long trips and short trips, which makes the energy consumption prediction more accurate.

5.3 Future Work

Firstly, the accuracy of speed prediction models needs to be improved further. Our current model mainly relies on the speed limit and the curvature of the road for prediction. But in the real-world situations, the vehicle speed is also related to the driver's driving behaviour, the line class of the road and the presence of traffic stops. In future work, we need to find a more suitable API that can provides more accurate information about the state of traffic and the type of road, including traffic signals, and take these features as inputs to train the model. Driver behaviour is also a significant factor in the speed prediction. A driver's behaviour can be judged and classified by collecting historical data such as acceleration and deceleration of the vehicle at different points in time, the force applied during braking and its duration, as well as the angle of steering wheel rotation and its rate of change. Based on different driver behaviours, we can train different models to make the speed prediction more accurate.

The other side is in the energy consumption prediction model. In this thesis we use data mostly from the Swedish winter season, when outside temperatures are low. When choosing the parameters we mainly consider the contribution of the resistance energy consumption, but we do not consider the contribution of the auxiliary energy consumption, which can be estimated based on the temperature setting in

the vehicle. Therefore, in future work, the test vehicles should record cabin temperature as an example of various auxiliary system energy consumption. This includes tracking interior temperature and air conditioning settings to enrich the features of our dataset. In addition, when the outdoor temperature gradually increases, we need to integrate the data collected by the vehicle at this time into the existing dataset to make the trained model more generalizable.

Besides, our model was trained on a single fixed-model vehicle, and the predictions may not work as well if the model applied to other models of Zeekr vehicles. Therefore, we need to deploy the data processing module on different types of test vehicles, so that the original model's dataset can be expanded and supplemented even when data is collected after each drive, and each model can be trained for each vehicle. This approach allows for a continuous flow of new data to train the model, which improves the accuracy of the model. However, this is not the only strategy. In scenarios where there is a large fleet of one model and limited data on newer vehicles, domain adaptation or transfer learning can be employed to adapt the existing model to new conditions effectively. Additionally, the vehicle properties may change over time and our model needs to be deployed within an MLOps (machine learning operations) setup that can handle deviations and use them to trigger model retraining when appropriate. This setup should monitor deviations in vehicle performance to trigger model retraining when it crosses a pre-determined threshold. Such a system ensures the model remains robust and accurate over time, adapting continuously to evolving conditions.

Bibliography

[1] N. Rauh, T. Franke, and J. F. Krems, "Understanding the impact of electric vehicle driving experience on range anxiety," *Human factors*, vol. 57, no. 1, pp. 177–187, 2015.

[2] A. Skuza and R. Jurecki, "Analysis of factors affecting the energy consumption of an ev vehicle-a literature study," in *IOP Conference Series: Materials Science and Engineering*, IOP Publishing, vol. 1247, 2022, p. 012 001.

[3] S. Barcellona, S. Grillo, and L. Piegari, "A simple battery model for ev range prediction: Theory and experimental validation," in *2016 International Conference on Electrical Systems for Aircraft, Railway, Ship Propulsion and Road Vehicles & International Transportation Electrification Conference (ESARS-ITEC)*, IEEE, 2016, pp. 1–7.

[4] H. Schmidt, "Worldwide harmonized light-vehicles test procedure (wltp) und real driving emissions (rde)–aktueller stand der diskussion und erste messergebnisse," pp. 1403–1411, 2015.

[5] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, "Electric vehicles energy consumption measurement and estimation," *Transportation Research Part D: Transport and Environment*, vol. 34, pp. 52–67, 2015.

[6] I. Miri, A. Fotouhi, and N. Ewin, "Electric vehicle energy consumption modelling and estimationa case study," *International Journal of Energy Research*, vol. 45, no. 1, pp. 501–520, 2021.

[7] C. De Cauwer, W. Verbeke, T. Coosemans, S. Faid, and J. Van Mierlo, "A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions," *Energies*, vol. 10, no. 5, p. 608, 2017.

[8] X. Qi, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Data-driven decomposition analysis and estimation of link-level electric vehicle energy consumption under real-world traffic conditions," *Transportation Research Part D: Transport and Environment*, vol. 64, pp. 36–52, 2018.

[9] S. Arsene, I. Sebean, A. Certan, and G. Popa, "Influence resistance at advancing on fuel consumption for vehicles that use an internal source of energy," *Procedia-Social and Behavioral Sciences*, vol. 186, pp. 573–581, 2015.

[10] F. Feng, R. Lu, and C. Zhu, "A combined state of charge estimation method for lithium-ion batteries used in a wide ambient temperature range," *Energies*, vol. 7, no. 5, pp. 3004–3032, 2014.

[11] J. Mamala, M. Graba, J. Mitrovic, K. Pranowski, and P. Stasiak, "Analysis of speed limit and energy consumption in electric vehicles," *Combustion Engines*, vol. 62, 2023.

[12] G. K. Tso and K. K. Yau, "Predicting electricity energy consumption: A comparison of regression analysis, decision tree and neural networks," *Energy*, vol. 32, no. 9, pp. 1761–1768, 2007, ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2006.11.010. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0360544206003288.

[13] L. Breiman, *Classification and regression trees*. Routledge, 2017.

[14] T. M. Ingason. "Insight into long short-term memory." (2021), [Online]. Available: https://thorirmar.com/post/insight_into_lstm/.

[15] J. Lee Rodgers and W. A. Nicewander, "Thirteen ways to look at the correlation coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[16] Y.-C. Chen, "A tutorial on kernel density estimation and recent advances," *Biostatistics & Epidemiology*, vol. 1, no. 1, pp. 161–187, 2017.

[17] S. Wglarczyk, "Kernel density estimation and its application," in *ITM web of conferences*, EDP Sciences, vol. 23, 2018, p. 00 037.

[18] R. M. Obrien, "A caution regarding rules of thumb for variance inflation factors," *Quality & quantity*, vol. 41, pp. 673–690, 2007.

[19] J. Ziegmann, J. Shi, T. Schnörer, and C. Endisch, "Analysis of individual driver velocity prediction using data-driven driver models with environmental features," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, IEEE, 2017, pp. 517–522.