Thesis for The Degree of Licentiate of Engineering

Predicting Electric Vehicle Energy Consumption from Field Data Using Machine Learning

Qingbo Zhu

Department of Architecture and Civil Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2025

Predicting Electric Vehicle Energy Consumption from Field Data Using Machine Learning

Qingbo Zhu

ISBN 978-91-8103-151-5
Doktorsavhandlingar vid Chalmers tekniska högskola, Ny serie nr 5609.
ISSN 0346-718X
Technical Report No. ACE 2025:2

Department of Architecture and Civil Engineering
Division of Geology and Geotechnics
Chalmers University of Technology
SE-412 96 Göteborg,
Sweden
Phone: +46(0)31 772 1000

Printed by Chalmers Digitaltryck,
Gothenburg, Sweden 2025.

To all the people who have helped and cared for me along the way

Abstract

This study addresses the challenge of accurately forecasting the energy consumption of electric vehicles (EVs), which is crucial for reducing range anxiety and advancing strategies for charging and energy optimization. Despite the limitations of current forecasting methods, including empirical, physics-based, and data-driven models, this paper presents a novel machine learning-based prediction framework. It integrates physics-informed features and combines offline global models with vehicle-specific online adaptation to enhance prediction accuracy and assess uncertainties. Our framework is tested extensively on data from a real-world fleet of EVs. The best global model, quantile regression neural network (QRNN), demonstrates an average error of 6.30%, the online adaptation further achieves a notable reduction to 5.04%, and both surpass the performance of existing models significantly. Moreover, for a 95% prediction interval, the online adapted QRNN improves coverage probability to 91.3% and reduces the average width of prediction intervals to 0.51 kWh. These results demonstrate the effectiveness and efficiency of utilizing physics-based features and vehicle-based online adaptation for predicting EV energy consumption.

Keywords

Electric vehicles, energy consumption, modeling and prediction, machine learning, field data.

This thesis is based on the following publication:

Qingbo Zhu, Yicun Huang, Chih Feng Lee, Peng Liu, Jin Zhang, Torsten Wik, *Predicting electric vehicle energy consumption from field data using machine learning.*
IEEE Transactions on Transportation Electrification (June 2024).
DOI: 10.1109/TTE.2024.3416532.

Acknowledgment

First and foremost, I would like to express my deepest gratitude to my supervisors, Prof. Torsten Wik and Dr. Yicun Huang. Thank you for stepping in during the middle of my PhD journey, taking on responsibilities that were not initially yours, and doing so selflessly without expecting anything in return. I would not have been able to continue my research without your unwavering support and encouragement. Your knowledge, experience, and guidance have been crucial to my research. Without you, this paper would not have been published, and I would not have had the opportunity to begin my licentiate seminar.

I would also like to thank my managers, Prof. Lars Rosén, Prof. Minna Karstunen, and Prof. Henriette Söderberg, as well as my director of studies, Dr. Rasmus Rempling. Your support, standing by my side, helped me find new hope and embark on a new journey during the darkest times in both my academic and personal life.

I am deeply grateful to my parents for sacrificing their own lives to take on my responsibilities and helping care for my young son for three years when I needed it most. I also thank my husband for his constant support in all my decisions, never once complaining. He is always by my side, accepting my emotions and mistakes, while also helping me find solutions and accomplish what I set out to achieve, helping me become a more optimistic, confident, and emotionally stable person. A special thanks to my son - I never asked if you wanted me as your mother, yet you have been the best son. It is my dream to be your role model, and that dream has kept me moving forward, never giving up.

Finally, I would like to thank everyone who has helped me along the way.

Acronyms

EV:	Electric vehicle
ETB:	Electric transit bus
SoC:	State of charge
NCM:	Nickel cobalt manganese oxide
HVAC:	Heating ventilation and air conditioning
GPS:	Global positioning system
ML:	Machine learning
PI:	Prediction interval
QR:	Quantile regression
QRNN:	Quantile regression neural network
QEGBR:	Quantile extreme gradient boosting regression
QRF:	Quantile regression forest
NN:	Neural network
DNN:	Deep neural network
XGBoost:	Extreme gradient boosting
RF:	Random forest
RMSE:	Root mean squared error
PMAE:	Percentage mean absolute error
CP:	Coverage probability
AW:	Average width

Contents

Part I

Summary

Chapter 1

Introduction

1.1 Motivation

Climate change is an urgent global crisis, largely driven by increased greenhouse gas emissions. According to the Intergovernmental Panel on Climate Change (IPCC), global surface temperatures have increased by approximately 1.1°C since the late 19th century, primarily due to human activities such as the burning of fossil fuels [1]. Carbon dioxide (CO2) remains the largest contributor to global warming, accounting for about 76% of total greenhouse gas emissions, followed by methane (16%) and nitrous oxide (6%) [2]. The continued rise in these emissions has led to unprecedented changes in weather patterns, sea level rise, and biodiversity loss. The United Nations Environment Programme's Emissions Gap Report estimates that current policies will result in a global temperature rise of 2.7°C by the end of the century, far above the 1.5°C target set in the Paris Agreement [3]. To mitigate these impacts, rapid reductions in greenhouse gas emissions are crucial across all sectors of the economy.

Current road transport, heavily relying on fossil fuels, has caused severe public concerns over the energy crisis, air pollution, and global warming. To achieve a sustainable transport system, the mass deployment of electric vehicles (EVs) is imperative and has become an unstoppable trend [4]. According to the International Energy Agency, the global EV stock in the stated policies scenario will expand rapidly from almost 18 million in 2021 to 200 million by 2030, corresponding to an average annual growth of more than 30% [5]. Such electric revolution in the transport sector entails various studies at different levels, ranging from vehicle components (e.g., batteries), individual EVs, and a vehicle fleet, up to traffic networks and their interactions with road infrastructure, power grids, and the environment [6], [7]. Specifically, typical research topics around EVs include but are not limited to battery sizing [8], charging planning [9], driving range prediction [10], routing [11], speed control [12], [13], energy optimization [14], and environmental analysis [15]. To tackle these problems, a common and fundamental task is the development of a reliable and accurate model for EV energy consumption. In addition, such an energy consumption model is a basis for making EV regulations and policies, and for analyzing the supply risks of battery resources.

Accurately and quickly predicting the energy consumption of EVs in completing a given trip is a non-trivial task due to the presence of several technical challenges. First, the energy of an EV is consumed by various resistances (e.g., caused by road friction, gravity, and aerodynamics), inevitable energy losses (e.g., in motors, batteries, and braking systems), and auxiliary vehicle components (e.g., heating, ventilation, and air conditioning system) while maintaining desired vehicle dynamics and comfort. Furthermore, this process involves a large set of parameters in vehicle design, operation, road topology, traffic states, and the external environment [16], some of which, such as the road conditions, wind speed, and driver behavior, are time-varying and stochastic. Compared to commercial transit buses, private electric cars tend to have complicated and highly volatile trips, and their prediction problem is even more challenging. Last but not least, an instantaneous prediction value is often expected for decision-making and system control, and, in contrast, a trip duration can range from several minutes to hours in which the associated energy consumption is related to vehicle dynamics varying in milliseconds. The multiple timescales involved further complicate the prediction task.

1.2 Contributions

The main contributions of this work can be summarized as:

- Proposing a new procedure to process and clean real-world EV data
- Constructing a comprehensive, physics-informed feature pool and extracting the best set of features
- Applying several powerful machine learning methods to develop prediction models for the consumed energy of an EV fleet with highly diverse trip information
- Providing the uncertainty range estimation associated with the point prediction, making it useful for corrective actions, decision-making, and safety control purposes
- Online adaptation of the selected global models for further improved accuracy and tightened uncertainty range

The proposed machine-learning pipeline is illustrated in Fig. 1.1.

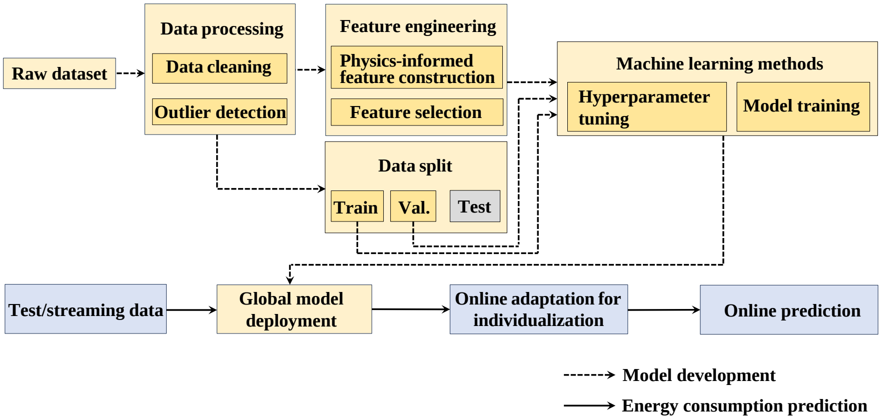

Figure 1.1: Pipeline to develop data-driven algorithms for energy consumption prediction.

1.3 Thesis outline

The thesis is organized as follows.

- Chapter 1 provides an introduction and motivation for the research work.
- Chapter 2 describes the dataset and the data-cleaning strategies.
- Chapter 3 briefly presents the physics-based modeling and the essential elements included.
- Chapter 4 details the model development process, including feature extraction, offline model training, and online model updating.
- Chapter 5 discusses the energy prediction results for the proposed model.
- Chapter 6 concludes the thesis and suggests potential future research directions.
- Appendix provides a table of the notation used.

Chapter 2

Data Description and Processing

2.1 Dataset

The dataset was collected by the National Monitoring and Management Platform for New Energy Vehicles in China from 55 battery electric taxis of the same brand and model. Equipped with lithium nickel cobalt manganese oxide (NCM) batteries having 30.4 kWh nominal capacity, these taxis were designed to have a driving range of up to 200 km and a maximum speed of 125 km/h. More specifications about them can be found in [17]. The states of vehicle operation (e.g., mileage and speed), batteries, motors, motor controllers, the braking system, fault alarms, and insulation resistance, as well as the location information, were monitored in real-time. The corresponding data were sent to the platform via wireless communication at a nominal frequency of 1 Hz. The detailed data items, formats, and communication protocols follow the standard given in [18]. The selected data reflect vehicles running in Beijing, with the earliest data points from March 2017 and the latest from December 2018.

The obtained time-series data for each vehicle were segmented into different driving trips, where the trips end whenever a stop or idling state is longer than three minutes. In practice, these trips could be terminated due to parking or charging. When there was no data uploading, embodied as data missing, or a series of zeros for more than three minutes, we also considered it as a trip completion. The data samples between every two consecutive trips were ignored as the corresponding energy consumption was negligibly small. Without energy consumption information from the vehicle cabin, e.g., for heating, ventilation, and air conditioning (HVAC), the energy output from the motors has been considered as the system output of interest, y, and was calculated by the time integral of the product of the measured current and voltage over the motor.

The measured GPS data from vehicles were input into Google Earth to generate road elevation data allowing calculation of road grade profiles. According to the recorded time and position, the external environment measurements, including ambient temperature, wind speed and direction, dewpoint temperature, and humidity, were taken from the weather website (https://www.xihe-energy.com) at a sampling time of 30 minutes.

All the trips were labeled by dates, and when the daily driving ranges became outside of $[1, 600]$ km, the corresponding trips were dropped. When the daily driving ranges are less than 1 km, the included trips are very short, rendering the energy consumption prediction unnecessary. On the other hand, it is unusual for these taxis to drive 600 km in a day as it means three full charges. As a result, a total of 91,932 trips were extracted from the raw vehicle data. Fig. 2.1 exemplifies the time-series velocity, acceleration, and elevation within two trips and illustrates the distribution of their trip-level average values over all the trips. It can be seen that the driving profiles vary largely within a specific trip and among different trips.

2.2 Data Processing

The dataset described in Chapter 2.1 was transmitted wirelessly from the running taxis to the data platform. However, wireless transmission is susceptible to interference and can be affected by long distances, physical obstructions, channel disturbance, and weather conditions. In addition, digital-to-analog conversion, sensor noise, and differentiation of measurements may also cause problems. Under such circumstances, our dataset should have inherently suffered from issues, such as measurement noise, data latency, loss, or mismatch. Given that data quality is critical for machine learning, the existing issues will inevitably weaken and even undermine the accuracy and reliability of data-driven models for predicting vehicle energy consumption. Hence, it is necessary to process and clean the data. To do so, we use knowledge of vehicle usage and kinetics as well as statistical properties to phase out potential issues.

1) Data Cleaning. The expected number of data samples serves as a reference for the ideal quantity points expected within a given time segment. For instance, assuming a sampling frequency of 1Hz and analysing a 10-second segment, the expected number of data samples, denoted as Σ_i, would be 10 under conditions of no data loss.

For any given trip $i \in \{1, 2, \cdots, \bar{N}\}$, where $\bar{N}$ is the number of all trips, we identify and quantify the data loss by comparing the number of existing time-series data samples, denoted by M_i, to its expected number of data Σ_i. The data loss rate, $\rho_{\text{loss},i}$, is defined as

$$\rho_{\text{loss},i} = \left(\Sigma_i - M_i\right)/\Sigma_i. \tag{2.1}$$

With the frequency of 1 Hz in collecting data, it is common that there exists $\rho_{\text{loss},i} > 0$. Analogously, the data mismatch rate, $\rho_{\text{mismatch},i}$, is calculated by

$$\rho_{\text{mismatch},i} = \left(M_i - M_{\text{mismatch},i}\right)/M_i, \tag{2.2}$$

where $M_{\text{mismatch},i}$ is the number of expected data samples and it can be exemplified as the situation where the motor current is zero while the vehicle speed is nonzero.

2) Outlier Detection. Some outliers in the obtained dataset can be detected from the labeled output while others can be isolated based on the trip-level features x that impact the system output. As illustrated in Fig. 2.2(a), some trips have consumed unusually more energy than others while a small number of trips are on the other side of the spectrum. From Fig. 2.2(b), it can be observed that the trip-level average energy consumption y_i tends to be linearly related to the driving distance d_i. In the bottom-left corner of the sub-figure, some trips are featured with over 10 km driving distance whereas the corresponding energy consumption is around zero. To systematically deal with these outliers, we combine the concepts of studentized residuals and leverage [19] to distinguish extreme output values in $\{y_1, \cdots, y_{\bar{N}}\}$ and extreme feature values in $\{d_1, \cdots, d_{\bar{N}}\}$.

Suppose $y_i = d_i\beta + \epsilon$ for the relationship between the energy consumption y_i and driving distance d_i, where β is a scalar coefficient and ϵ is the intercept, representing the slope and bias of the linear regression model, respectively. Then, according to the analytical solution to the linear least squares problem, the optimal value can be obtained as $[\beta, \epsilon]^T = (D^TD)^{-1}D^TY$ and $\hat{Y} = D(D^TD)^{-1}D^TY$, where $Y = [y_1, \cdots, y_{\bar{N}}]^T$ and $D = [d_1, 1; \cdots; d_{\bar{N}}, 1]$. By defining $D^\dagger$ as an intermediate matrix, specifically $D^\dagger = D(D^TD)^{-1}D^T$, which is valid under the assumption that D^TD is invertible, the estimated value of Y is given by

$$\hat{Y} = \begin{bmatrix} D^\dagger_{11} & D^\dagger_{12} & \cdots & D^\dagger_{1\bar{N}} \\ D^\dagger_{21} & D^\dagger_{22} & \cdots & D^\dagger_{2\bar{N}} \\ \vdots & \vdots & \ddots & \vdots \\ D^\dagger_{\bar{N}1} & D^\dagger_{\bar{N}2} & \cdots & D^\dagger_{\bar{N}\bar{N}} \end{bmatrix} Y, \tag{2.3}$$

where $D^\dagger_{ii}$ is the leverage value and it indicates the distance between a certain driving distance d_i and the average value of d_i for all the $\bar{N}$ trips. When $D^\dagger_{ii}$ exceeds a predefined threshold, the corresponding i-th data point is flagged and identified as an outlier.

According to its definition, the studentized residual for trip i, denoted by r_i, is given by [19]

$$r_i = \frac{y_i - \hat{y}_i}{\sqrt{\sigma(y_1, \cdots, y_{\bar{N}}) \cdot (1 - D^\dagger_{ii})}}, \tag{2.4}$$

where $\hat{y}_i$ is derived from (2.3), $y_i - \hat{y}_i$ represents the ordinary residual for trip i, and the function $\sigma(\cdot)$ is the standard deviation of y_i in all the $\bar{N}$ trips. An observation i with an internally studentized residual larger than 3 is generally deemed an outlier.

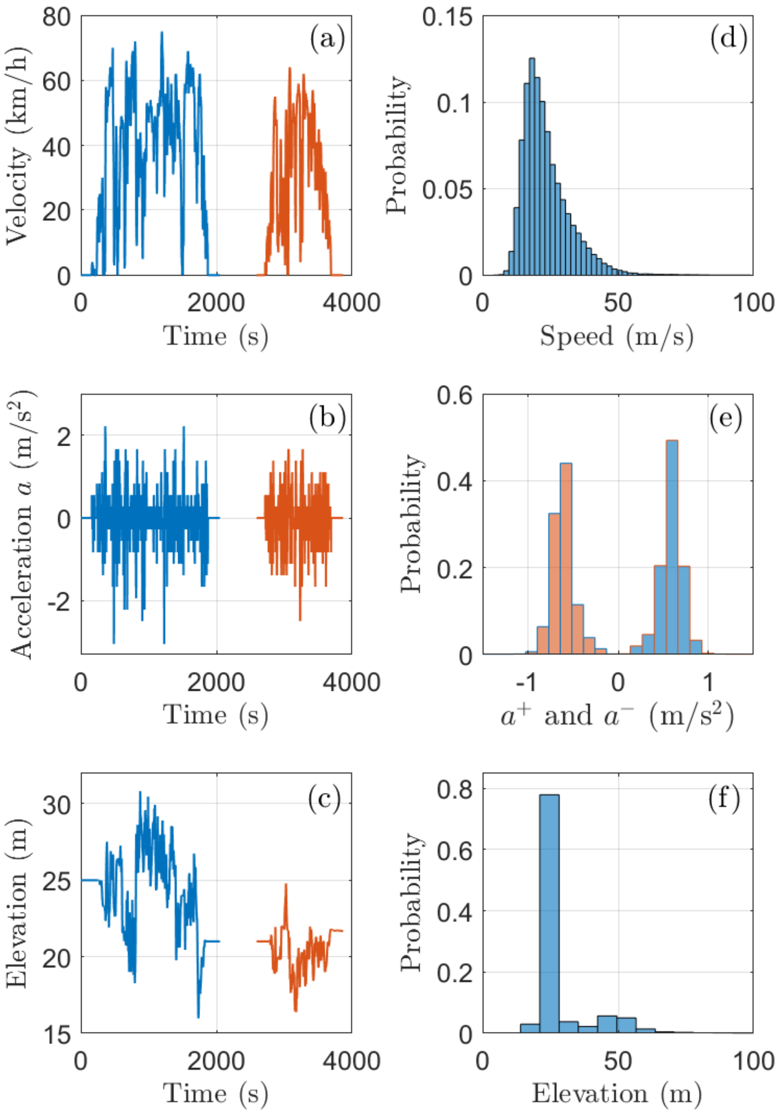

Figure 2.1: Data measured from electric taxis. (a)-(c) show the velocity, acceleration, and elevation profiles, respectively, of two continuous trips of a vehicle. (d)-(f) illustrate the histogram trip-level average velocity, acceleration, and elevation, respectively, over all the trips, where both the positive acceleration, a^+, and the negative acceleration, a^-, are considered.

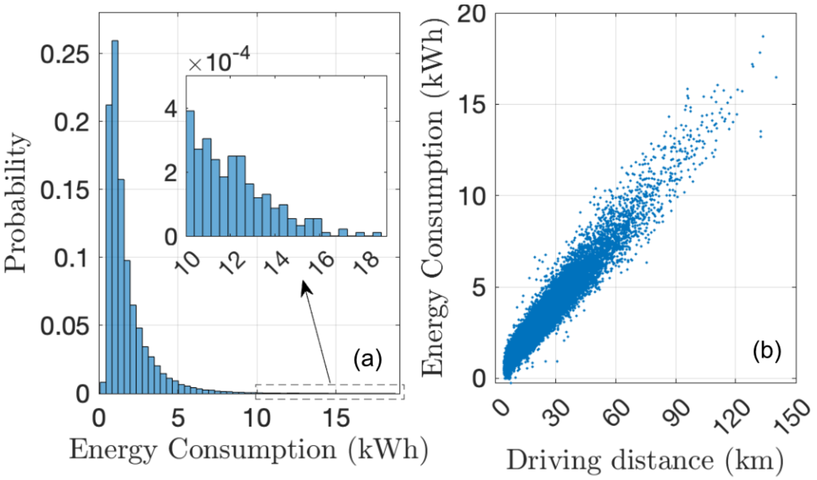

Figure 2.2: Illustration of raw data samples. (a) Probability of energy consumption over all the trips. (b) Total energy consumption of each trip versus the corresponding driving distance.

Chapter 3

Overview of Physics-based Modeling

In general, the energy consumption of a vehicle is to overcome several types of driving resistance and to support auxiliary systems, e.g., HVAC. At the same time, it will be affected by regenerative braking and energy efficiencies in the powertrain system and its components. By only considering energy flow from the vehicle motors, the energy consumption of an electric vehicle, $\hat{y}$, can be calculated by

$$\hat{y} = F_r d/\eta_r + \eta_b E_{\text{brake}}, \tag{3.1}$$

where F_r represents the propulsion force, E_{brake} is the regenerative braking energy (negative), and η_r and η_b denote the corresponding energy efficiencies. According to Newton's second law of motion, the propulsion force applied to vehicles can be expressed as [20]

$$\begin{aligned} F_r = {} & mgf_r\cos(\theta) + \frac{C_D A(V - V_{\text{air}})^2}{21.15} \\ & + mg\sin(\theta) + \delta ma, \end{aligned} \tag{3.2}$$

where the four terms on the right-hand side of (3.2) represent the rolling resistance, air resistance, climb resistance, and acceleration resistance. g, δ, and C_D denote the gravitational acceleration, the transfer coefficient from the revolving mass to a linear mass, and the drag coefficient, respectively, and these three parameters are generally constant during vehicle movement. m, A, and f_r are the vehicle mass, equivalent cross-sectional area, and tire rolling resistance coefficient. While m can vary among different trips for a taxi, f_r and A are heavily influenced by road conditions and the ambient environment. θ, V, and V_{air} are the road grade, vehicle velocity, and wind velocity, which are variables in a trip.

If all the model parameters in (3.2) are known *a priori* and all the variables can be measured accurately, the instantaneous propulsion force can be calculated directly at each time step. However, as noted in the introduction, these parameters and a set of energy efficiencies for a vehicle system can be affected by many complicated factors, and it is very expensive and difficult to quantify them accurately, particularly considering the wide range of uncertainties and stochasticity during vehicle usage.

Chapter 4

Data-driven model development

Fully recognizing the complexities involved in precise parameterization and long-term simulation of physical models, this study adopts a data-driven approach to efficiently forecast the trip-level energy consumption of EVs. This approach incorporates a set of features carefully constructed from the physics-based model described in Chapter 3.

4.1 Physics-informed Feature Construction and Engineering

The first step in establishing a machine learning (ML) model is to extract elements for feature construction. All these elements are taken directly or are inspired by the physics-based model (3.2). As such, all essential physical insights into energy consumption can be systematically incorporated into the ML-based prediction model. As shown in the first three columns of Table 4.1, these elements can be categorized into four classes, i.e., trip intrinsic attributes, road characteristics, vehicle states, and ambient environments. Note that in addition to the instantaneous acceleration, the positive and negative values of acceleration are considered to better reflect the vehicle states on the trip level. The relative wind velocity V_w, defined as $V - V_{\text{air}}$, is also considered as part of the ambient environment.

With these physics-informed elements, a two-step procedure is used to construct a comprehensive feature pool. First, the time-series data of each of the considered elements over a given trip is transformed into a form of histogram. Then, a variety of statistical properties of the histogram can be extracted, including the mean, variance, 0.95 quantile, and 0.05 quantile, which represent the characteristics of central tendency, dispersion, and extreme situations of each driving trip, respectively. The obtained library of elements and corresponding constructed features are listed in Table 4.1. Note that in the absence of measured data, our feature pool does not explicitly incorporate locally distributed traffic information, such as traffic density and congestion levels. However, we anticipate that the selected features related to vehicle velocity and acceleration implicitly capture the effects of varying traffic conditions on trip-based energy consumption.

This employed feature construction strategy compresses hundreds or thousands of time-series data samples in a trip into a small number of features, corresponding to each physical element. The strategy reduces the scale of the input data by several orders of magnitude, resulting in significantly decreased memory resources to store the data and computational cost to train ML models. In addition, it enables efficient predictions during the online implementation. Such a strategy is imperative when the raw data are stored originally and locally as histograms within the vehicle. A similar strategy was used in [21] to compress vehicle field data for predicting the aging trajectory of lithium-ion batteries.

With the constructed feature pool, feature engineering is conducted to select a set of most relevant and independent features for the development of ML models. To select the most relevant features, Spearman correlation analysis is first conducted to assess the correlation between any feature x and the system output y.This method evaluates the strength and direction of their monotonic relationship, providing a robust measure even for non-linear dependencies [22]. To quantify this relationship, the Spearman's rank correlation coefficient ρ_s can be calculated as

$$\Delta_i = R_{x_i} - R_{y_i}, \tag{4.1}$$

$$\rho_s = 1 - \frac{6\Sigma_{i=1}^{N_\text{train}} \Delta_i^2}{N_\text{train}(N_\text{train}^2 - 1)}, \tag{4.2}$$

where i is the index of the trip-based data samples, N_train represents the number of training samples, and R_{x_i} signifies the rank of x_i after sorting all training samples for the considered feature in ascending order. By setting a lower threshold for the Spearman's rank correlation coefficient, i.e., $\rho_{s,\min}$, any features with a score less than $\rho_{s,\min}$ will be discarded from the feature pool. The remaining features in the feature pool are those that exhibit a strong monotonic relationship with the system output, ensuring their relevance to the target variable.

Following the analysis for the correlation between any feature and the system output, Pearson correlation analysis [23] is applied to quantify the correlation between any two features within the remaining pool. Pearson correlation measures the strength and direction of the linear relationship between two variables, with values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). It is particularly useful for identifying redundancy between features. Unlike statistical significance testing, which often assumes normality of the data, Pearson correlation itself quantifies the linear dependency between features regardless of their individual distributions [24]. For each feature pair, if the Pearson correlation coefficient ρ_p exceeds a specified upper threshold $\rho_{p,\max}$, the feature with a lower ρ_s value (indicating less relevance to the target variable) is abandoned to avoid multicollinearity among features.

Table 4.1: Elements for feature construction and the resulting feature pool

Element classification	Elements	Description	Features
Trip intrinsic attributes	d	Driving distance of a trip	d
	t_d	Driving time of a trip	t_d
Road characteristics	E	Elevation of the road	mean, variance, 0.95 quantile, 0.05 quantile
	$G^{\cos}$	Cosine value of road grade	mean, variance, 0.95 quantile, 0.05 quantile
	$G^{\tan}$	Tangent value of road grade	mean, variance, 0.95 quantile, 0.05 quantile
Vehicle states	V	Vehicle velocity	mean, variance, 0.95 quantile, 0.05 quantile
	V^2	Square of vehicle velocity	mean, variance, 0.95 quantile, 0.05 quantile
	V^3	Cube of vehicle velocity	mean, variance, 0.95 quantile, 0.05 quantile
	a	Acceleration of vehicle	variance, 0.95 quantile, 0.05 quantile
	a^+	Positive vehicle acceleration	mean
	a^-	Negative vehicle acceleration	mean
Ambient environments	V_w	Relative wind velocity	mean, variance, 0.95 quantile, 0.05 quantile
	V_w^2	Square of relative wind velocity	mean, variance, 0.95 quantile, 0.05 quantile
	T_a	Ambient temperature	mean
	T_d	Dewpoint temperature	mean
	H	Humidity	mean
	P	Precipitation	mean

4.2 Prediction Intervals

A prediction interval (PI) is a statistical range within which a future observation is expected to fall with a specified level of confidence, typically 95% or 99% [25]. Its main function is to capture the uncertainty around a prediction by incorporating both model error and data variability, offering a range that reflects the natural variability and uncertainty inherent in the model and data [26].

The physical interpretation of a prediction interval is straightforward: it provides a range $[L, U]$ where a new observation Y, given $X = x$, is expected to fall with a specified confidence level. Here, L represents the lower bound, U the upper bound, and the interval $[L, U]$ satisfies the condition: $P(L \leq Y \leq U) = \text{Confidence Level}$.

Prediction intervals can be constructed using quantiles. For instance, A 95% prediction interval for the value Y is given by

$$I(x) = [Q_{.025}(x), Q_{.975}(x)]. \tag{4.3}$$

In this context, $Q_{.025}(x)$ represents the 2.5th percentile of the conditional distribution of Y given $X = x$, and $Q_{.975}(x)$ denotes the 97.5th percentile. Together, these quantiles define a range that is expected to contain the true value Y with 95% confidence, meaning Y has a 95% probability of falling within the interval $I(x)$.

The width of this prediction interval can vary greatly with x. For some values of x, the prediction intervals may be significantly narrower, indicating that predictions can be made with much greater accuracy for those specific cases. This implies that the uncertainty around the prediction is lower, and the model can confidently predict the outcome with a smaller margin of error. Conversely, for other values of x, the prediction intervals may be much wider, reflecting higher uncertainty and less reliable predictions.

This variability in the width of prediction intervals is particularly noticeable in certain data sets, where the relationship between the predictor variables and the outcome is not uniform. Factors such as heteroscedasticity (where the variance of errors differs across levels of the predictor) or complex patterns in the data can lead to significant differences in prediction accuracy across the range of x.

Quantile regression addresses this issue effectively by directly estimating the conditional quantiles of the response variable, rather than assuming a constant variance or a specific error distribution. This allows for the construction of prediction intervals that adapt to the changing uncertainty across different values of x. As a result, quantile regression provides a principled and flexible approach to assessing the reliability of predictions, particularly in cases where traditional methods may fall short.

4.3 ML-based Prediction Models

The effectiveness of machine learning across various domains largely depends on the choice of algorithm, as each algorithm possesses distinct strengths and is suited to particular types of problems. Machine learning techniques can be broadly categorized into supervised, unsupervised, semi-supervised, and reinforcement learning. These categories represent a wide spectrum of methods developed to address various data-driven problems, including classification, regression, clustering, dimensionality reduction, and anomaly detection.

Within the realm of supervised machine learning, regression methods can be broadly categorized based on their characteristics and applications. For example, traditional linear regression methods are widely used for their simplicity and interoperability. More complex non-linear approaches, like decision trees and their ensemble methods, can capture intricate patterns in data, while neural networks are often utilized for their flexibility and capability to model highly non-linear relationships. Additionally, probabilistic regression models, such as Gaussian Process Regression, provide estimates of uncertainty by assuming specific output distributions [27]. In our study, trip-based EV energy consumption prediction is framed as a regression problem in terms of the selected features and the consumed energy y. The target is to develop reliable and accurate ML models that provide trip-wise point predictions and the associated uncertainty range, where the latter aims to make the prediction results interpretable. Given the diverse range of available regression methods, four algorithms are selected for our prediction target, specifically, quantile regression (QR), quantile regression neural network (QRNN), quantile extreme gradient boosting regression (QEGBR), and quantile regression forests (QRF), are utilized in the offline pathway of Fig. 1.1 to develop novel prediction models for y. Note that none of these quantile-based algorithms relies on the assumption of any specific distribution of the system output, unlike other probabilistic models, such as Gaussian process regression.

4.3.1 QR

QR is a straightforward learning algorithm for estimating the conditional quantiles of the target variable y based on observed data, extending beyond the median. Unlike classical linear regression, which focuses on estimating the conditional mean of the output given the input variables, QR provides a more comprehensive analysis of the conditional distribution of y. Linear regression is effective when standard assumptions-such as normality, and homoscedasticity of residuals-are met. In contrast, QR offers greater flexibility by not relying on these assumptions.

From a physical perspective, QR divides the training dataset into two segments based on the value of quantile hyperparameters. While QR utilizes the same regression equation as linear regression, its distinctive predictions are revealed through its objective function, commonly referred to as the check function. The QR's check function, $\rho_\tau(e_i, \tau)$, can be written as [28]

$$\rho_\tau(e_i, \tau) = \begin{cases} -(1-\tau)e_i & \text{if } e_i < 0 \\ \tau e_i & \text{if } e_i \geq 0, \end{cases} \tag{4.4}$$

where the parameter $\tau \in (0, 1)$ represents the quantile of y, and the discrepancy between y_i and its predicted value $\hat{y}_i$ is defined as $e_i = y_i - \hat{y}_i$ for the i-th trip. If $\tau = 0.5$, the goal is to fit a straight line that divides the dataset into two equal parts, which is equivalent to using the absolute loss (AL), $\mathcal{L}_{AL}(y_i, \hat{y}_i) = \sum_{i=1}^{N_{\text{train}}} |y_i - \hat{y}_i|$, as the loss function in linear regression. The determination of the upper and lower bounds for a specified prediction interval can be achieved by setting the values of τ in (4.4).

The QR's loss function is presented in Fig. 4.1. This figure illustrates the loss incurred for different quantiles, τ, by asymmetrically weighting the residuals. Each curve in the figure corresponds to a specific quantile value, showing how the check function penalizes positive and negative discrepancies differently, depending on the chosen quantile. This allows QR to capture the conditional distribution of outputs more effectively by adjusting the loss function to different data portions.

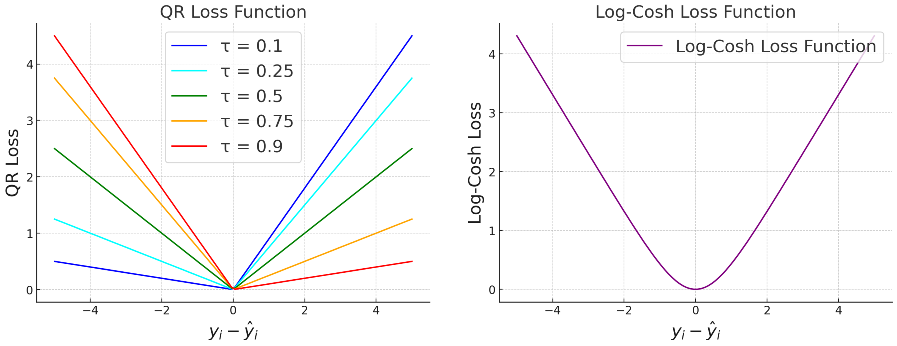

Figure 4.1: Comparison of QR and Log-Cosh Loss Functions

4.3.2 QEGBR

While QR offers simplicity and interpretability, it may struggle with capturing complex, non-linear relationships in the data. This is where tree-based algorithms come into play, offering a powerful alternative by modeling decision paths and capturing intricate patterns within the data. Classification and Regression Trees (CARTs) are the foundational tree-based algorithms that utilize a single tree structure to address classification and regression tasks. They recursively partition the dataset into subsets, selecting the most significant feature at each node to guide the splits. This results in a tree where each branch corresponds to a decision path and the leaves signify the final prediction outcomes [29]. Due to the limited predictive power and the tendency of single decision trees to overfit, ensemble methods have been introduced to enhance overall model performance. In the context of tree-based methods for regression, two primary ensemble strategies have been developed to improve accuracy and robustness: bagging and boosting.

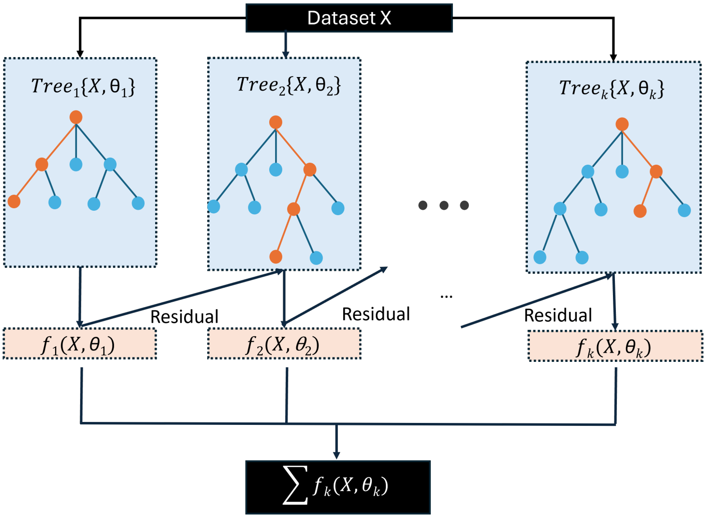

Figure 4.2: Structure visualization of XGBoost

Extreme gradient boosting (XGBoost), whose structure is shown in Fig. 4.2, is a scalable tree-boosting algorithm that builds a series of decision trees to minimize prediction errors sequentially [30]. It begins with a baseline prediction, typically the mean of the target values in the training data. Each subsequent tree is trained to predict the residuals of the previous trees. By iteratively adjusting the model in this way, XGBoost refines its predictions, reducing the discrepancy between the predicted and actual values. During each iteration, a new tree is added to focus on correcting the residuals left over by the ensemble of previous trees. The final prediction is obtained by summing the predictions from all the trees, resulting in a model that combines these weak learners into a strong predictor.

While CARTs do not rely on a loss function to make predictions, XGBoost is fundamentally based on a specified loss function for optimization. Unlike QR, which allows for flexibility in selecting loss functions that only require first-order differentiability, XGBoost mandates that its loss function be second-order differentiable. This requirement is necessary to support Newton's method for efficient optimization, as it leverages both the first and second derivatives in the process [31].

Newton's method is advantageous in XGBoost because it provides a powerful approach for optimizing the objective function by incorporating both the gradient and curvature of the loss function. In each boosting iteration, XGBoost updates its predictions by adding a new tree to the model, effectively minimizing the objective function with respect to this tree. This objective function consists of two components: the prediction error and a regularization term that penalizes model complexity, preventing overfitting.

Specifically, the regularized objective function $\mathcal{L}_{\text{XG}}$ is composed of the sum of the prediction errors overall training samples and a regularization term that controls the complexity of the model, denoted as follows:

$$\mathcal{L}_{\text{XG}} = \sum_{i=1}^{N_{\text{train}}} \ell_L(y_i, \hat{y_i}) + \sum_{k=1}^{T} \Omega(f_k). \tag{4.5}$$

In this function, $\ell_L(y_i, \hat{y_i})$ is a differentiable convex loss function measuring the discrepancy between the true labels y_i and the predictions $\hat{y_i}$, summed over all N_{train} samples in the training dataset. Additionally, $\sum_{k=1}^{T} \Omega(f_k)$ represents the regularization term, where T is the total number of trees in the model, and k indexes each individual tree. The regularization term $\Omega(f_k)$ penalizes the complexity of each tree f_k, encouraging the model to remain simple by limiting the depth and number of leaves. The tree ensemble model in (4.5) comprises functions as parameters, making it incompatible with traditional optimization techniques in Euclidean space. Instead, the model is trained through an additive approach. Formally, with $\hat{y_i}^{(t)}$ denoting the prediction for the i-th instance at the t-th iteration, and f_t is added to minimize the following objective function.

$$\mathcal{L}_{\text{XG}}^{(t)} = \sum_{i=1}^{N_{\text{train}}} \ell_L(y_i, \hat{y_i}^{(t-1)} + f_t(x_i)) + \Omega(f_t). \tag{4.6}$$

Its second-order approximation is given by

$$\mathcal{L}_{\text{XG}}^{(t)} \approx \sum_{i=1}^{N_{\text{train}}} [\ell_L(y_i, \hat{y}^{(t-1)}) + \mathbf{g}_i f_t(x_i) + \frac{1}{2}\mathbf{h}_i f_t^2(x_i)] + \Omega(f_t). \tag{4.7}$$

This can be used to quickly optimize the objective in the general setting [32]. Here, $\mathbf{g}_i$ and $\mathbf{h}_i$ represent the gradient and Hessian (first and second derivatives) of the loss function with respect to y_i. These terms allow XGBoost to update predictions more accurately and with fewer iterations, optimizing both the speed and precision of the training process. By using the second-order derivative information, XGBoost constructs each new tree in a way that minimizes the objective more effectively, with the optimal weights for each tree's leaf node w_j calculated as

$$w_j = -\frac{\sum_{i\in j} \mathbf{g}_i}{\sum_{i\in j} \mathbf{h}_i + \lambda}. \tag{4.8}$$

In this equation, $\sum_{i\in j} \mathbf{g}_i$ and $\sum_{i\in j} \mathbf{h}_i$ represent the summed gradients and Hessians for all samples within the leaf j, and λ is a regularization parameter.

To satisfy the requirements for both first- and second-order gradient statistics in the loss function, XGBoost typically employs the log-cosh loss function, as illustrated in Fig. 4.1, which takes the following form

$$\mathcal{L}_{\mathrm{LC}}(y_i, \hat{y}_i) = \log(\cosh(\hat{y}_i - y_i)). \tag{4.9}$$

It helps XGBoost make point predictions but not capture uncertainty ranges. To have both the point and uncertainty range predictions, one would naturally expect some appropriate combinations of XGBoost and QR. Given that QR's loss function is not second-order differentiable at the origin, a second-order differentiable function can be introduced to create a smooth approximation of the QR loss function, allowing for a smooth transition at the origin [33]. With this in mind, we replace $\hat{y}_i - y_i$ in (4.9) by the quantile check function $\rho_\tau(\hat{y}_i - y_i, \tau)$ defined in (4.4). The resulting algorithm, which combines XGBoost with a new loss function, is referred to as QEGBR. The formulation of this new loss function is presented in Fig. 4.3.

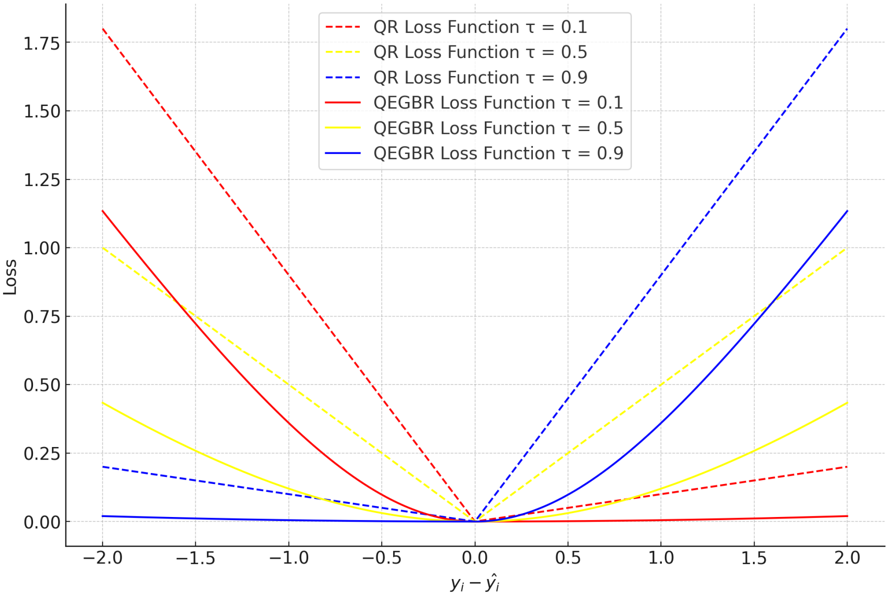

Figure 4.3: Visualization of QR and QEGBR Loss Functions

4.3.3 QRF

Another tree-based ensemble algorithm is Random Forests (RFs). As shown in Fig. 4.4, RFs is a scalable algorithm that uses bagging to build multiple decision trees in parallel on random subsets of data. Different from QR and QEGBR algorithms, which predict the target value by minimizing their respective loss functions, random forest-based algorithms are non-parametric tree-based approaches without the process of optimizing parameters. To extend RFs to

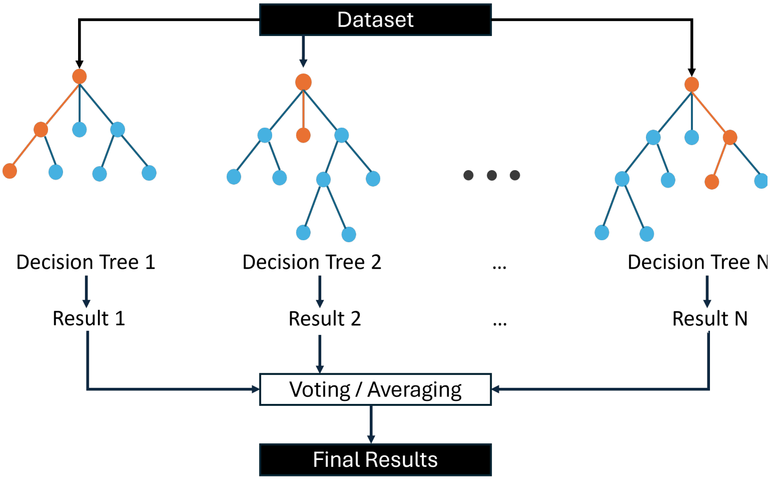

Figure 4.4: Visualization of Random Forest Architecture

accommodate uncertainty quantification, QRF was developed. QRF modifies the traditional RFs framework to predict conditional quantiles, providing a way to capture uncertainty ranges along with point predictions.[34]. When constructing the forest, both QRF and RF utilize decision trees, employ bootstrapping to generate distinct subsets of data, and make random selections for nodes and splitting points. In RF, the predicted value represents the conditional mean, which is approximated by averaging the predictions from all the trees in the forest. For N_{train} independent observations

$$(Y_i, X_i), i = 1, 2, ..., N_{train},$$

The prediction generated by a single tree , denoted by $\mathcal{T}(\phi)$, for a new data point $X = x$, is obtained by averaging the observed values within the corresponding leaf $\ell(x, \phi)$. Here, ϕ represents the random parameter vector that defines the tree's structure and its splitting rules. Let the weight vector $w_i(x, \phi)$ be assigned a positive constant if observation X_i is part of leaf $\ell(x, \phi)$ and 0 otherwise. Since the weights sum to one, the prediction can be expressed as

$$w_i(x, \phi) = \frac{1_{\{X_i \in R_\ell(x,\phi)\}}}{\#\{j : X_j \in R_\ell(x, \phi)\}}, \tag{4.10}$$

where $1_{\{X_i \in R_\ell(x,\phi)\}} = 1$ if $X_i \in R_\ell(x, \phi)$ and 0 otherwise. The prediction of a single tree, given covariate $X = x$, is then the weighted average of the original observations Y_i, $i = 1, 2, ..., n$, according to

$$\text{single tree: } \hat{\mu}(x) = \sum_{i=1}^{N_{\text{train}}} w_i(x, \phi) Y_i.$$

In RFs model, the conditional mean $E(Y|X = x)$ is approximated by averaging the predictions from k individual trees, each constructed using an independent and identically distributed (i.i.d) parameter vector ϕ_t for $t = 1, ..., k$. Let w_i represent the average of the weight over this ensemble of trees, defined as the mean weight assigned to observation i across the collection, i.e.

$$w_i(x, \phi) = k^{(-1)} \sum_{t=1}^{k} w_i(x_i, \phi_t). \tag{4.11}$$

The prediction of RFs is then

$$\text{RFs: } \hat{\mu}(x) = \sum_{i=1}^{N_{\text{train}}} w_i(x) Y_i.$$

The weighted observations provide not only a good approximation to the conditional mean but also to the full conditional distribution, forming an important foundation of QRF. The conditional distribution function of Y, given $X = x$, is given by

$$F(y|X = x) = P(Y \leq y|X = x) = E(1_{\{Y \leq y\}}|X = x)$$

Given that $E(Y|X = x)$ is approximated by a weighted mean over the observations of Y, an approximation to $E(1_{\{Y \leq y\}}|X = x)$ is defined as the weighted mean over the observations of $1_{\{Y \leq y\}}$,

$$\hat{F}(y|X = x) = \sum_{i=1}^{N_{train}} w_i(x) 1_{\{Y \leq y\}} \tag{4.12}$$

The algorithm for computing the estimate $\hat{F}(y|X = x)$ can be summarized as:

(a) Grow k trees $\mathcal{T}(\phi_t)$ for $t = 1, ..., k$, following the procedure of random forests. However, instead of recording only the average of observations in each leaf, retain all individual observations within each leaf for every tree.

(b) For a given $X = x$, pass x through all trees. Calculate the wight $w_i(x, \phi_t)$ of observation $i \in 1, ..., N_{train}$ for each tree as specified in (4.10). Then, determine the weight $w_i(x)$ for each observation i by taking the average of $w_i(x, \phi_t)$ across all trees, as described in (4.11).

(c) Using the weights from step(b), compute the estimate of the distribution function for all $y \in \mathbb{R}$ as specified in (4.12).

For a given value of quantile α, the conditional α-quantile $Q_\alpha(x)$, defined as $Q_\alpha(x) = inf\{y : F(y|X = x) \geq \alpha\}$, is estimated by substituting $\hat{F}(y|X = x)$ for $F(y|X = x)$ in the equation above, yielding the estimate $\hat{Q}_\alpha(x)$.

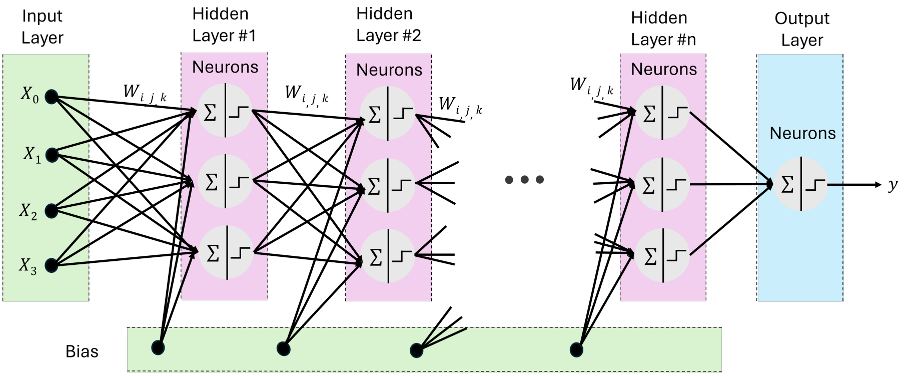

Figure 4.5: Visualization of DNNs

4.3.4 QRNN

Neural networks (NNs), drawing inspiration from the human brain, are computational models composed of interconnected nodes structured into layers. These nodes, or 'neurons,' are linked through weighted connections and apply activation functions to process data in a forward direction for prediction tasks [35]. The learning process, known as backpropagation, iteratively adjusts these weights, allowing the network to improve its predictions over time by minimizing error. NNs often include multiple hidden layers, enabling the development of deep learning models that can represent complex data patterns.

Deep neural networks (DNNs) are a type of neural network with multiple hidden layers, enabling them to learn intricate representations and handle complex datasets [36]. A typical DNN architecture, illustrated in Fig. 4.5, consists of an input layer, multiple hidden layers, and an output layer. Each neural in a given layer l_{layer} receives weighted inputs from the previous layer $(l_{\text{layer}} - 1)$, applies an activation function, and outputs the result to the next layer $(l_{\text{layer}} + 1)$. This can be represented as follows:

(a) Weighted Sum: Each neuron j in layer l_{layer} calculates a weighted sum of inputs from the preceding layer $(l_{\text{layer}} - 1)$:

$$z_j^{(l_{\text{layer}})} = \sum_i w_{ij}^{(l_{\text{layer}})} a_{act,i}^{(l_{\text{layer}}-1)} + b_j^{(l_{\text{layer}})}, \tag{4.13}$$

where $w_{ij}^{(l_{\text{layer}})}$ represents the weight connecting neuron i in layer $(l_{\text{layer}} - 1)$ to neuron j in layer l_{layer}, $a_{act,i}^{(l_{\text{layer}}-1)}$ denotes the activation of neuron i in the previous layer, and $b_j^{(l_{\text{layer}})}$ is the bias term for neuron j in layer l_{layer}.

(b) Each neuron's output $a_{act,j}^{l_{\text{layer}}}$ is computed by applying an activation function σ_{act} to $z_j^{l_{\text{layer}}}$:

$$a_{act,j}^{(l_{\text{layer}})} = \sigma_{act}(z_j^{(l_{\text{layer}})}) \tag{4.14}$$

where common choices for σ_{act} include ReLU, defined as $ReLU(z) = max(0, z)$, and the sigmoid function, $\sigma_{sig}(z) = \frac{1}{1+e^{-z}}$. These non-linear functions allow the network to capture complex, non-linear relationships within the data.

(c) Output layer: For a regression problem, the output layer may produce a single prediction $\hat{y}$ by linearly combining the final hidden layer's activations:

$$\hat{y} = \sum_j w_j^{(L_{layer}+1)} a_{\text{act},j}^{(L_{layer})} + b^{(L_{layer}+1)} \tag{4.15}$$

where L_{layer} represents the index of the last hidden layer. This linear combination allows the network to make continuous predictions suitable for regression problems.

Once the DNNs produce outputs (i.e., predictions for each sample) in the output layer, the loss function calculates the discrepancy between these predictions and the actual target values (labels) in the training data, serving as a measure of prediction accuracy. In this process, the network first generates predictions in the output layer for each sample in the batch. Then, these predictions are compared to the actual target values using a specified loss function. The computed loss value drives the backpropagation process, which iteratively adjusts the weights across all layers of the network based on the calculated gradients, gradually improving the network's predictive accuracy.

In DNNs for regression problems, the most commonly used loss functions are the squared loss (SL), $\mathcal{L}_{SL}(y_i, \hat{y}_i) = \sum_{i=1}^{N_{\text{train}}} (y_i - \hat{y}_i)^2$, and AL, but they are restricted to making point predictions. According to [37], by integrating the QR algorithm into a DNN structure, one can obtain both the point prediction $\hat{y}$ and the uncertainty range estimation associated with $\hat{y}$. In this work, we incorporate the QR check function into the loss function of QRNN, namely $\mathcal{L}(y_i, \hat{y}_i) = \sum_{i=1}^{N_{\text{train}}} \rho_\tau(e_i, \tau)$.

4.4 Online Model Adaptation for Customized Prediction

The global models in Chapter 4.3 are trained and validated on historical vehicle data. When we apply the resulting models to predict vehicle energy consumption, they are blind to the unique characteristics of new vehicles whose driving situations may deviate largely from those in the training set. The predictions are essentially generated from an open-loop simulation based on the global models. The historical driving data of a targeted vehicle during real-world usage shall contain valuable information for understanding and learning the characteristics of its future energy consumption. Taking this individualized information into consideration as feedback, online adaptive models can be developed to potentially improve the prediction performance, particularly for vehicles that have not been seen during the training process. To test this concept for EV fleets, we for the first time develop online adaptive models for EV energy consumption based on QRNN and QEGBR, where model adaptations are made in real-time based on the latest trip the targeted vehicle has completed.

4.4.1 Online Adaptive QRNN

With continuous usage of the targeted vehicle, new data samples will be available and can be used for individualized model development. The QRNN often consists of many layers and neurons to learn complex input-output relationships.

However, incorporating new data into the model presents a challenge. On one hand, if the entire QRNN model is re-trained at each time step k using only the latest data sample, it is likely to result in severe overfitting, as the model may become excessively tuned to the most recent sample and lose its generalizability. On the other hand, if new data samples are simply appended to the existing training set for periodic re-training, the process can be computationally expensive and time-consuming, making it impractical for real-time or online adaptation. Additionally, under this approach, the model's performance will likely remain dominated by the initially available offline dataset, limiting the influence of the newly acquired data.

To address these limitations and enable efficient online adaptation, we adopt the learning-without-forgetting approach proposed by [38]. The essence of this method is to preserve the knowledge encoded in the global model while selectively updating only a subset of parameters to incorporate new information. In this context, we maintain the overall structure and most parameters of the QRNN model, while specifically updating the parameters of the hidden layer closest to the output layer, denoted as Θ. This selective update ensures that the model retains its learned relationships while adapting to new data.

To facilitate online parameter estimation efficiently, we employ stochastic gradient descent (SGD) to iteratively update Θ with each new data sample. This process allows for recursive adaptation without full model re-training. For a new trip at time step $\kappa + 1$, the parameter Θ are updated according to the following rule:

$$\Theta_{\kappa+1} = \Theta_{\kappa} - \gamma \nabla \mathcal{L}(\Theta_{\kappa}), \tag{4.16}$$

where $\kappa + 1$ denotes the trip next after trip κ, α represents the learning rate, which controls the step size of each update, and $\nabla \mathcal{L}(\Theta_{\kappa})$ is the gradient (vector) of the QRNN loss function with respect to the parameter Θ_{κ}. By focusing on updating only the parameters closest to the output layer, this approach enhances the influence of new data on the model's predictions, thus enabling a balanced blend of historical knowledge and recent observations.

4.4.2 Online Adaptive QEGBR

XGBoost is widely recognized for its mature and well-encapsulated implementation, which generally limits modifications to its globally trained offline model.

The core of the QEGBR algorithm, adapted from XGBoost, involves adding trees to refine residuals from previous predictions, building a model that learns from its errors across multiple quantiles. To extend the benefits of QEGBR into a dynamic, real-time context, an Online Adaptive QEGBR algorithm is proposed, enabling continuous, individualized learning as new data becomes available. Specifically, with each new data point arriving from a target vehicle, an additional tree is appended to the QEGBR model, allowing it to refine predictions in real-time by reducing residuals based on the latest observations. This process ensures that the model remains current with each vehicle's unique characteristics without retraining the entire model from scratch, allowing it to quickly adapt to changing conditions. The online adaptive QEGBR alters the structure, rather than the parameters, of its corresponding global model.

Note that during the online phase, both the above two adaptation algorithms will preserve the major model information from the previous learning step to reduce the risk of over-fitting, robustness issues, and large modeling errors.

Chapter 5

Results and Discussion

Four evaluation metrics are applied to evaluate the prediction accuracy and uncertainty estimation performance. Two of them are used to analyze the prediction accuracy, namely the root mean squared error (RMSE) and the percentage mean absolute error (PMAE) defined as

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y_i})^2}, \tag{5.1}$$

$$PMAE = \frac{\frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y_i}|}{\frac{1}{N}\sum_{i=1}^{N}|y_i|} \times 100\%, \tag{5.2}$$

where N is the number of data samples in the testing set. The other two evaluation standards for assessing prediction intervals at the same probability level are the coverage probability (CP) and average width (AW) of the prediction interval, defined as

$$PI_{\mathrm{CP}} = \frac{1}{N}\sum_{i=1}^{N}C_i, \text{where } C_i = \begin{cases} 1 & y_i \in [\underline{y}_i, \overline{y}_i] \\ 0 & y_i \notin [\underline{y}_i, \overline{y}_i] \end{cases} \tag{5.3}$$

$$PI_{\mathrm{AW}} = \frac{1}{N}\sum_{i=1}^{N}(\overline{y}_i - \underline{y}_i), \tag{5.4}$$

where $\underline{y}_i$ and $\overline{y}_i$ represent the predicted lower and upper bounds, respectively, for a certain prediction interval.

5.1 Implementation Specification

For data processing in Chapter 2.2, to balance the degree of removing the detected issues and the number of data samples in model development, we set the tolerable thresholds for $\rho_{\mathrm{loss},i}$ and $\rho_{\mathrm{mismatch},i}$ to 10% and 30%, respectively. Any trips in the dataset that do not satisfy the conditions will be removed. We flag trip-level samples as outliers when $D^{\dagger}_{ii}$ exceeds $6/\bar{N}$ or the absolute value of r_i is larger than 3.

We use Monte Carlo cross-validation to evaluate the accuracy, efficiency, and robustness of the developed ML models. From the processed dataset, 45 vehicles are randomly selected and used for model training, while the data from the remaining 10 vehicles serve as the test set. This process of random data splitting is iterated 20 times to mitigate sample bias. Subsequently, the average results of these iterations are computed to provide a robust assessment of model performance.

For feature engineering in Chapter 4.1, we set $\rho_{s,\min} = 0.05$, and $\rho_{p,\max} = 0.8$.

For all the ML algorithms described in Chapter 4.3, the quantile values, τ, in their corresponding loss functions are set as 0.5 for generating point predictions. The quantile values are set to be 0.025 and 0.975 to obtain the lower and upper bounds, of a 95% prediction interval, respectively.

5.2 Results of Data Processing

The results of data processing, including data cleaning and outlier detection conducted in Chapter 2.1, are partially depicted. For brevity, only the data on trip-based energy consumption and its correlation with driving distances are presented in Fig. 5.1.

Compared with the raw data samples displayed in Fig. 2.2, the shapes of these two histogram plots are highly similar, both exhibiting a distinct right skewness. This means most trips had low energy consumption. However, the distribution of the cleaned dataset has a shorter and lighter tail than that of the raw dataset. Specifically, the trips with energy consumption of more than 15 kWh, which is about 50% of the maximum energy stored in the battery system at the beginning of its life, are largely reduced. The removed data samples particularly include "skeptical trips" that consumed high energy within short driving distances. After the data processing, the linearity between driving distances and energy consumption becomes more pronounced, and correspondingly, the variation of energy consumption generally becomes smaller for a given driving distance.

In total, 91,932 driving trips are extracted from the original time-series data. According to the tolerable thresholds of data loss and mismatch in each trip specified in Chapter 5.1, 53.88% of the data samples are removed. After the data cleaning, 816 data samples belong to the defined outliers, corresponding to 0.8876% of the raw data. Overall, 41,585 samples are retained for the training and testing of ML models.

5.3 Results of Feature Engineering

This subsection presents results of feature engineering performed in Chapter 4.1 using $\rho_{s,\min}$ and $\rho_{p,\max}$ specified in Chapter 5.1. Table 5.1 lists the 17 selected features and their coefficients of Spearman correlation with the output y, i.e.,

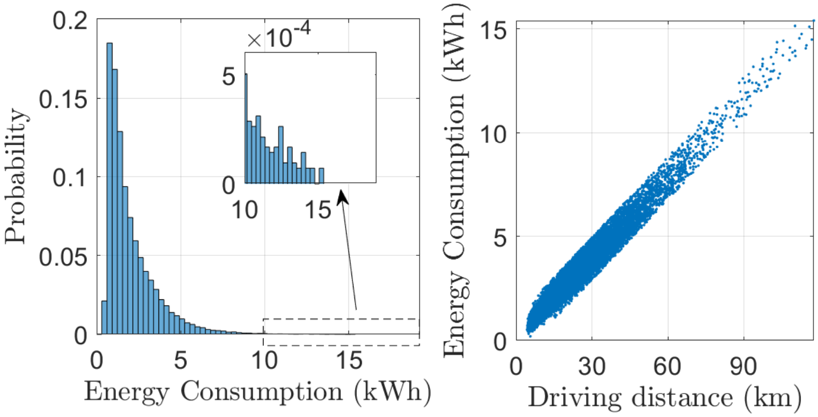

Figure 5.1: Illustration of the trip-based energy consumption data resulting from data cleaning and outlier detection.

the trip-based energy consumption. For Pearson correlation between each two features, the coefficients are also derived, and Fig. 5.2. depicts the scores for the features applied in the ML models.

The driving distance d is found as the most relevant feature to predict y, having a Spearman correlation coefficient as high as 0.98. However, several other features also correlate strongly with y, such as the driving time. However, these features are heavily dependent on d according to the Pearson correlation analysis and were therefore excluded to mitigate multicollinearity. In general, features related to vehicle states are more relevant to y than those features associated with the ambient environment. Specifically, the 95th quantile of the vehicle velocity carries more weight than the average velocity, and the variance of acceleration takes precedence over all other acceleration-related features. It is noteworthy that the impact of both the variance of the relative wind velocity and the elevation on y is substantial, which is an underexplored aspect in the existing literature.

5.4 Results of the Global ML Models

5.4.1 Prediction Accuracy

By using the ML-based prediction models developed in Chapter 4.3, we can continuously predict EV energy consumption in each trip. With the data split defined in Chapter 5.1 and the obtained hyperparameters in Table 5.2, the overall prediction results for all data samples in the test set are summarized in Tables 5.4–5.3.

Although the task is challenging, it can be observed that the best model, i.e., QRNN, can accurately predict EV energy consumption. Specifically, after cleaning the data and removing the outliers, the QRNN model can deliver

Table 5.1: The selected features and their Spearman correlation coefficients

Feature	Description	Coefficient
d	Driving distance	0.9831
$V_{w,\text{var}}$	Variance of relative wind velocity	0.4126
E_{var}	Variance of elevation	0.3928
V_{95}	95th quantile of vehicle velocity	0.2945
V_{ave}	Average of vehicle velocity	0.2422
a_{var}	Variance of acceleration	0.2126
E_{95}	95th quantile of of elevation	0.1871
$G^{\cos}_{\text{var}}$	Variance of $G^{\cos}$	0.1576
$G^{\cos}_{95}$	95th quantile of $G^{\cos}$	0.1399
T_d	Average of dewpoint temperature	0.1313
P	Mean precipitation	-0.0944
H	Mean humidity	0.0880
$V_{w,5}$	5th quantile of relative wind velocity	-0.0879
$G^{\tan}_{5}$	5th quantile of $G^{\tan}$	0.0752
V_5	5th quantile of vehicle velocity	-0.0665
$G^{\tan}_{95}$	95th quantile of $G^{\tan}$	-0.0618
$G^{\tan}_{\text{var}}$	Variance of $G^{\tan}$	0.0586

predictions with a PMAE of 6.3%. In addition to QRNN, QEGBR can also effectively learn the characteristics of energy consumption from the diverse field data and make reliable predictions for any given input that has not been seen during training. These results verify the effectiveness of the developed energy consumption models as well as the constructed and selected physics-informed features underlying each model.

To evaluate the efficacy of our newly developed global models for energy prediction, as well as their underlying feature set (i.e., those in Table 5.1 and labeled here as Set 4), we compare the obtained results with those achieved in three benchmarks using different feature sets. The first benchmark has a feature set (i.e., Set 1) only incorporating driving distance. The second benchmark adopts a more complex feature set derived in the state-of-the-art literature [17], which includes driving range, driving time, average velocity, 95% quantile of acceleration, 5% quantile of acceleration, and average temperature, and six categorical variables representing traffic conditions during rush and non-rush hours across various time frames and days. The third benchmark employs the proposed feature engineering but only takes the ten features with the highest correlation coefficients from Table 5.1, forming Set 3. To ensure a fair comparison, all four machine learning (ML) models were implemented and fine-tuned across each benchmark. The results demonstrate a clear advantage of

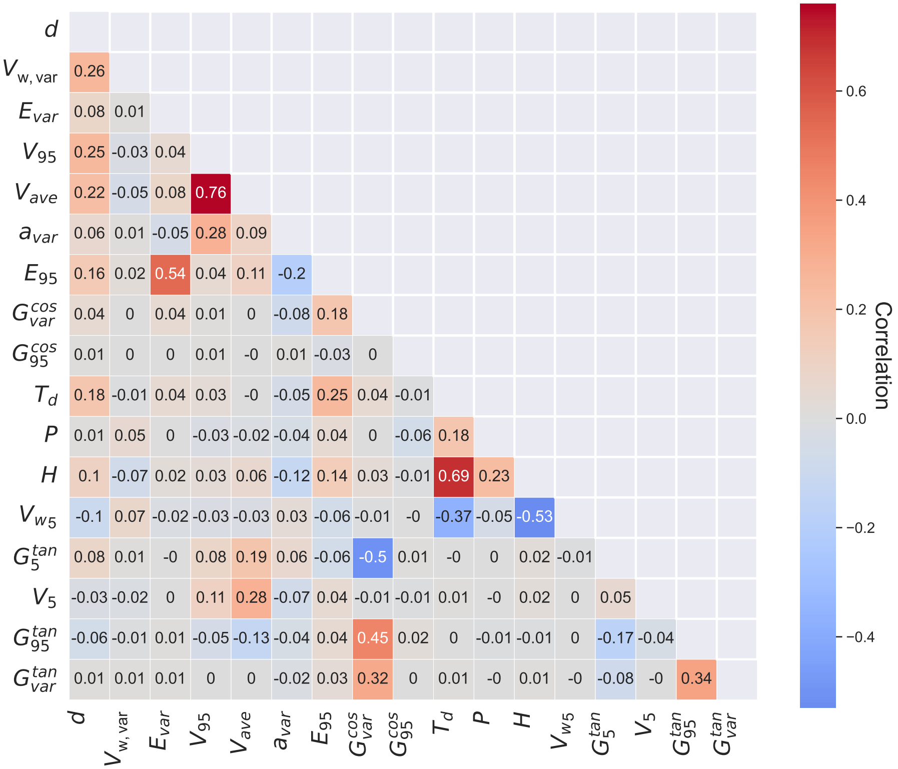

Figure 5.2: Pearson correlation heatmap of the selected features for the processed dataset.

using our comprehensive feature set (Set 4). Notably, our QRNN model achieves a reduction in prediction error of 11.9% compared to the best-performing benchmark and 19.6% relative to the state-of-the-art benchmark.

To assess the impact of data processing on prediction accuracy, we conducted a comparative analysis of results using the raw dataset and two processed datasets. Our findings indicate significant improvements in prediction accuracy after data cleaning and outlier removal, as evidenced by reductions in both the RMSE and PMAE. By using QRNN as an example, without the proposed data processing techniques, the PMAE can become 41% larger, and the RMSE will increase by 29%. Analysis of the results presented in the last two columns of Table 5.4 underscores that in addition to addressing data loss and mismatch issues, it is crucial to remove outliers.

Upon detailed examination of the results, it is observed that for the two less accurate models, i.e., QR and QRF, the RMSE values exhibit a marginal increase after the data cleaning, a phenomenon that initially appears counter-intuitive. It is found that the prediction errors from QR and QRF models tend to escalate for longer trips. The data cleaning process primarily removes data

Table 5.2: Hyperparameter values for the applied ML algorithms

QR	—
QRNN	Hidden unit numbers: 256, 64, 32, 32, 8 L2 regularization: 0.0005 Learning rate: 0.00001
QEGBR	Learning rate: 0.01 Total number of iterations: 8000 Early stopping rounds: 10
QRF	Maximum features: 17 Maximum tree depth: 20 Minimum sample split: 10 Minimum sample leaf: 2

Table 5.3: Effects of different feature sets and online learning on prediction accuracy

Model	Error type	Global models with different features				Online adaptive models
		Set 1	Set 2	Set 3	Set 4	Set 4
QR	RMSE	0.2891	0.2365	0.2343	0.2263	╲
	PMAE	10.37%	8.29%	8.17%	7.90%	
QRF	RMSE	0.2868	0.2232	0.2266	0.2132	╲
	PMAE	10.26%	7.88%	7.92%	7.31%	
QRNN	RMSE	0.2870	0.2193	0.2129	0.1788	0.1456
	PMAE	10.30%	7.84%	7.23%	6.30%	5.04%
QEGBR	RMSE	0.2875	0.2120	0.2012	0.1861	0.1604
	PMAE	10.28%	7.50%	6.97%	6.47%	5.56%

samples pertaining to shorter trips, which results in an increase in the average trip distance within the cleaned dataset. Consequently, this leads to slightly increased RMSE values, specifically 0.274 for the QR model and 0.2506 for the QRF model. This observation further corroborates the significance of outlier removal from the dataset.

By using the processed dataset, the prediction results for individual vehicles in the test set are illustrated in Fig. 5.3. QRNN and QEGBR generally outperform the other two models for individual vehicles and trips, which is consistent with the results obtained above. It can also be seen that the prediction errors do not appreciably increase with energy consumption (Fig. 5.3c), showing the stability and robustness of the developed models. This implies that for trips with higher energy consumption, the relative errors tend to be smaller. In Fig. 5.3a–b, the trajectories of the four ML models have a similar variation trend. This conveys that in addition to the ML algorithms, the prediction results are also influenced by other factors, such as the data quality in terms of

Table 5.4: Prediction errors of the developed models using different datasets

Model	Error type	Raw data	Data with outliers	Fully processed data
QR	RMSE	0.2694	0.2740	0.2263
	PMAE	10.23%	8.43%	7.90%
QRF	RMSE	0.2482	0.2506	0.2132
	PMAE	9.54%	7.70%	7.31%
QRNN	RMSE	0.2311	0.2262	0.1788
	PMAE	8.88%	7.00%	6.30%
QEGBR	RMSE	0.2247	0.2128	0.1861
	PMAE	8.84%	6.74%	6.47%

resolution and level of detail. It may be noticed that for test vehicle No. 6, the predictions have a low RMSE but a high PMAE. This is because we have used the average energy consumption of all its trips, and shorter distances traveled by this vehicle result in a larger PMAE value.

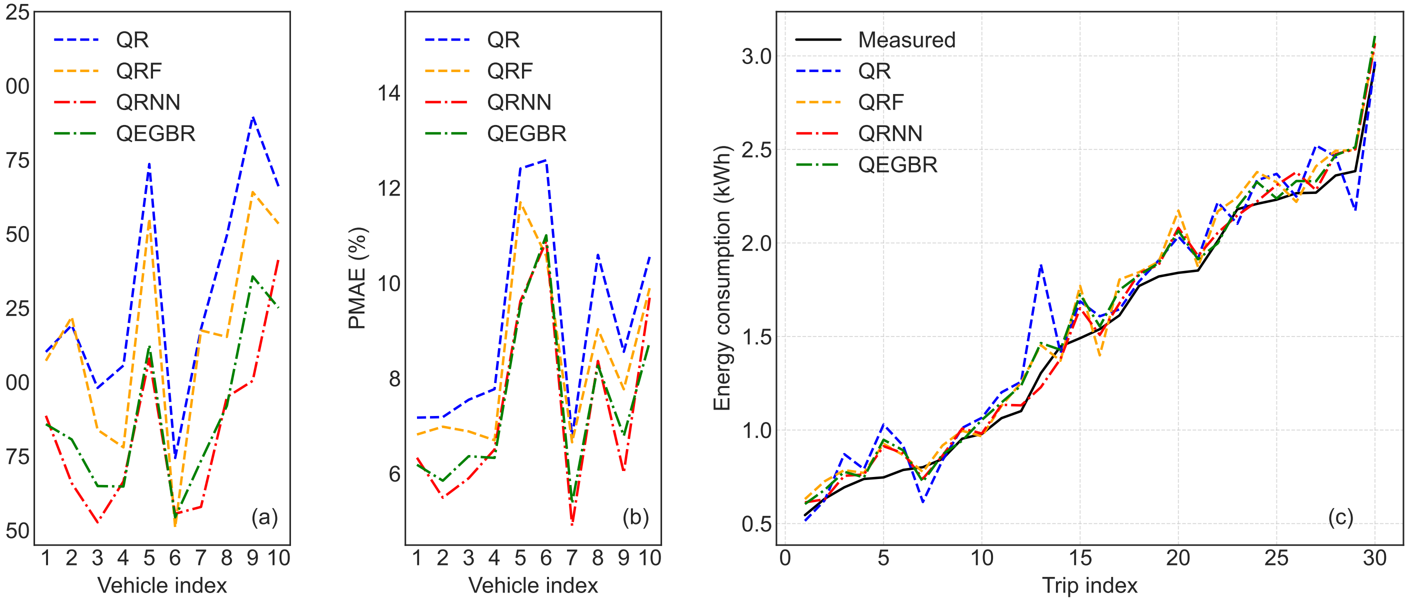

Figure 5.3: Prediction results of the four global models for individual vehicles. (a) and (b) show the RMSE and PMAE values. (c) exemplifies the predicted energy consumption in 30 trips from a randomly selected vehicle in the test set.

5.4.2 Uncertainty Estimation

To make the predictions interpretable for decision-making of EV charging and energy usage, four quantile-based ML algorithms have been used in the prediction model development allowing the uncertainty associated with each prediction to be estimated at the same time. To the best of our knowledge, this has not previously been conducted in the literature of data-driven EV energy prediction. The coverage probability and average width of prediction intervals, i.e., $\mathrm{PI}_{\mathrm{CP}}$ and $\mathrm{PI}_{\mathrm{AW}}$, are used to quantify the performance of uncertainty estimation, with the results presented in Table 5.5 and Fig. 5.4. Without doubt, one would desire prediction intervals to always cover the ground truth (i.e., high $\mathrm{PI}_{\mathrm{CP}}$) and to be as narrow as possible (namely low $\mathrm{PI}_{\mathrm{AW}}$). Within a 95% confidence interval, the ideal $\mathrm{PI}_{\mathrm{CP}}$ for models evaluated on the test dataset is 0.95, though the actual values of $\mathrm{PI}_{\mathrm{CP}}$ may vary with the data distribution of the test dataset and model structures.

From the numerical and graphical results, it can be seen that the prediction intervals generated by QR, QRF, and QRNN are able to cover the measured trip-level energy consumption on most occasions. While QR gives the highest $\mathrm{PI}_{\mathrm{CP}}$, QEGBR results in lowest $\mathrm{PI}_{\mathrm{AW}}$ thanks to the use of the synthetic quantile loss function, i.e., $\rho_{LC}(y_i, \hat{y}_i) = \log(\cosh(\rho_\tau(\hat{y}_i - y_i, \tau))$ as described in Chapter 4.3.2. QRNN is capable of best balancing $\mathrm{PI}_{\mathrm{CP}}$ and $\mathrm{PI}_{\mathrm{AW}}$, and can consequently serve as the most suitable candidate for uncertainty estimation. By leveraging QRNN's prediction interval bounds, i.e., $\underline{y}_i$ and $\overline{y}_i$ generated for each trip i, it is possible to establish suitable constraints and safety margins for various decision actions.

5.4.3 Computational efficiency

In addition to performance for point prediction and uncertainty estimation, the computational efficiency of ML models is crucial for real-time implementations. With this consideration, we investigate the computational time required by all the developed models. It is found that the most accurate global model, namely the QRNN, requires only 15 microseconds on average to predict the energy consumption for individual trips. This time is significantly less than the trip duration, rendering it negligible.

Table 5.5: Uncertainty estimation by the global models and online adaptive models

Evaluation matrices	QR	QRF	QRNN	QEGBR	Online QRNN	Online QEGBR
$\mathrm{PI}_{\mathrm{CP}}$	0.9354	0.9165	0.8931	0.6377	0.9127	0.5785
$\mathrm{PI}_{\mathrm{AW}}$	0.7446	0.6624	0.5981	0.5294	0.5082	0.4348

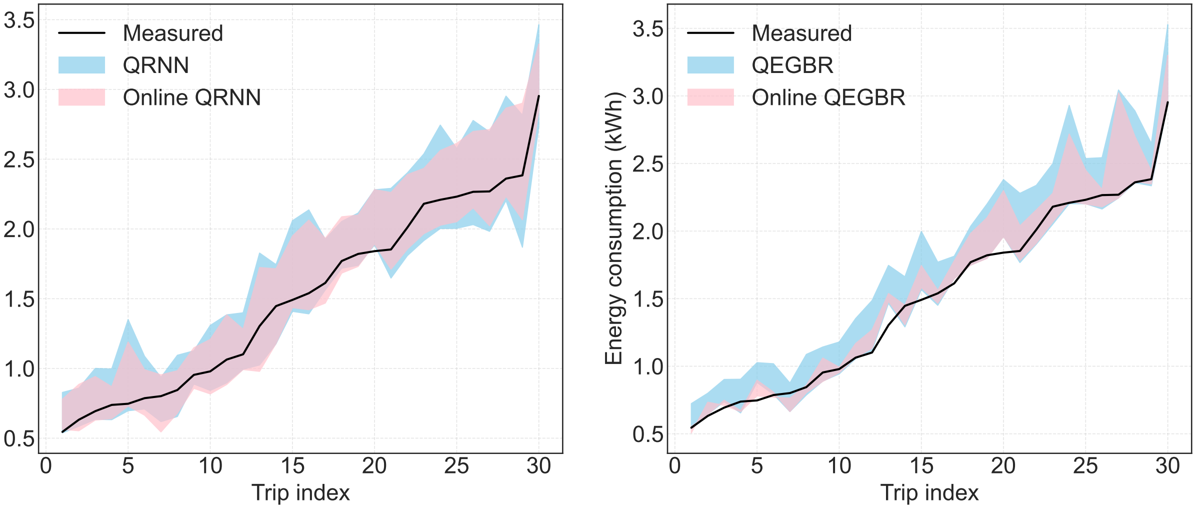

Figure 5.4: The estimated uncertainties and the ground truth of energy consumption for a vehicle randomly selected from the test set, where the rose- and turquoise-colored areas denote the 95% prediction intervals.

5.5 Results of Online Adaptive Models

The best global models, i.e., QRNN and QEGBR, have been adapted online, with the results demonstrated in Table 5.4 and Fig. 5.5. It is evident that the two online adaptive ML models significantly outperform their global models across all vehicles in the test set (see Fig. 5.5a). The online QRNN can deliver the highest accuracy, with a PMAE of 5.04%. Corresponding to a reduction of more than 20% compared to the offline model performance. In comparison with the method proposed in [17], the reduction is as high as 35%. QEGBR, with a PMAE of 5.56%, is also superior to all the global models. Further, for individual trips of a randomly selected vehicle (see Fig. 5.5b–c), the predicted values of the online adapted QRNN and QEGBR closely follow the ground truth in the entire range of energy consumption. This validates that our proposed online adaptation method can judiciously learn the energy consumption behavior of the target EV and effectively combine it with the corresponding global model.

The effect of online adaption on uncertainty estimation is also investigated. From Table 5.5, it is clear that the online QRNN and QEGBR effectively reduce the average width of prediction intervals, $\mathrm{PI}_{\mathrm{AW}}$, showing a decrease of 15% and 18%, respectively. Similar results are observed in Fig. 5.4 for individual trips. Obviously, the turquoise-colored areas of the online adaptive models are smaller than the rose-colored areas of the global models. Furthermore, the online QRNN cannot only tighten the prediction intervals but also increase the probability of containing the measured trajectory inside the intervals. This makes its estimated bounds of each prediction interval, i.e., $\underline{y}$ and $\overline{y}$, more valuable for advanced EV energy management. By contrast, the online QEGBR shrinks $\mathrm{PI}_{\mathrm{AW}}$ but also decreases $\mathrm{PI}_{\mathrm{CP}}$. Such a low value of $\mathrm{PI}_{\mathrm{CP}}$ implies that the uncertainty predictions are less reliable and useful.

While significantly enhancing prediction and estimation performance, the online model adaptation will inevitably demand additional computational effort. For instance, when using the online QRNN model, the average computational time is 4.73 milliseconds for predicting the energy consumption of a single trip. This minimal duration makes the developed ML models highly suitable for online vehicular applications.

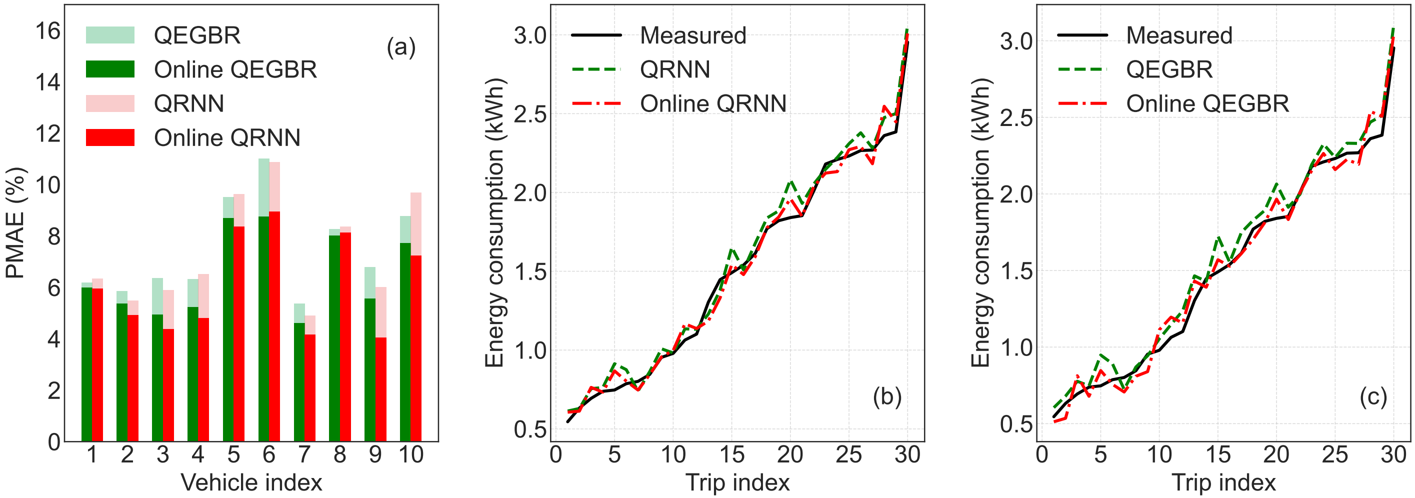

Figure 5.5: Comparison of the point prediction results of the global models (QRNN and QEGBR) and their online adaptive models. (a) PMAE values for all vehicles in the test set. (b) and (c) Trajectories of the ground truth and predictions for 30 trips from a randomly selected vehicle in the test set.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This research has introduced a field data-based ML pipeline for the prediction of EV energy consumption. The first novelty arises from the proposed data processing method tailored for a large amount of real-world EV data that was inherently plagued by various issues and outliers. Then, a new feature set was constructed from physical insights and picked meticulously through systematic correlation analyses. Based on these data and features, four quantile-based machine learning algorithms were pertinently formulated and innovatively applied for the EV energy prediction, enabling accurate and reliable prediction of both the energy consumption and associated uncertainties. Finally, the best-performing global ML models were adapted online for individualized predictions, leading to consistently improved accuracy and tightened confidence internals.

The developed ML models, as well as their underpinning data processing and feature engineering, were validated extensively for EV energy prediction. Comprehensive comparisons were conducted for different steps of data processing, between global models and online adaptive models, and with models in the literature. The online adaptive QRNN models outperformed all other models with an average prediction error of 5.04%, corresponding to an over 35% improvement over the state-of-the-art models. Substantial advantages have also been observed from different steps of data processing and online model adaptation.

6.2 Future Work

This thesis has established a solid foundation for predicting motor energy consumption in EVs. Future research could build on this work in these areas, including but not limited to:

- Battery Energy Consumption: This study focuses solely on predicting motor energy consumption due to the limited availability of auxiliary data. Extending this work to predict total battery energy consumption, accounting for auxiliary systems, would provide a more comprehensive understanding of energy consumption of EVs.

- Data Expansion: By removing 1.9% of the data that deviated significantly from the main data trend, this model achieved a 16% improvement in RMSE and a 13% improvement in PMAE. Expanding the training dataset with additional data could allow the model to learn from these types of outliers more sufficiently, ultimately leading to a more comprehensive and robust predictive model.

Appendix

Table 6.1: Lowercase Forms and Variants of Symbols

Symbol	Meaning	Unit
a	Acceleration of vehicle	m/s^2
a^+	Positive vehicle acceleration	m/s^2
a^-	Negative vehicle acceleration	m/s^2
a_{act}	Activation	-
b	Bias term	-
d	Driving distance of a trip	km
e	Prediction error	-
f	Independent tree structure	-
f_r	Tire rolling resistance coefficient	-
g	Gravitational acceleration	-
$\mathbf{g}$	Gradient of the loss function	-
$\mathbf{h}$	Hessian of the loss function	-
k	Individual tree index	-
ℓ	A leaf in random forests	-
ℓ_L	Differentiable convex loss function	-
m	Vehicle mass	-
r	Studentized residual	-
t_d	Driving time of a trip	h
w	Weight for a leaf node	-
y	Observed energy consumption	-
$\hat{y}$	Predicted energy consumption	-
z	Weighted sum	-

Table 6.2: Capital Forms and Variants of Symbols

Symbol	Meaning	Unit
A	Equivalent cross-sectional area	-
C_D	Drag coefficient	-

$D^{\dagger}$	Leverage value	-
E	Elevation of the road	m
E_{brake}	Regenerative braking	-
F_r	Propulsion force	-
G^{cos}	Cosine value of the road grade	-
G^{tan}	Tangent value of the road grade	-
H	Humidity	g/m^3
I	Prediction interval	-
L	Lower bound	-
L_{layer}	Last hidden layer	-
$\mathcal{L}$	Loss function	-
$\mathcal{L}_{AL}$	Absolute loss function	-
$\mathcal{L}_{SL}$	Squared loss function	-
$\mathcal{L}_{XG}$	XGBoost loss function	-
M	The existing number of data samples	-
N	The number of test samples	-
$\bar{N}$	The number of all samples	-
N_{train}	The number of training samples	-
P	Precipitation	mm
Q_{α}	α-th quantile	-
R_x	The rank of x	-
T	Total number of trees	-
T_a	Ambient temperature	°C
T_d	Dewpoint temperature	°C
$\mathcal{T}$	Prediction of a single tree	-
U	Upper bound	-
V	Vehicle velocity	km/h
V_{air}	Wind velocity	-
V_w	Relative wind velocity	km/h
X	Predictor variable	-
Y	Real-valued response variable	-

Table 6.3: Greek Letter Forms and Variants of Symbols

Symbol	Meaning	Unit
β	Scalar coefficient	-
γ	Learning rate	-
δ	Transfer coefficient	-

ϵ	Intercept	-
η_b	Energy efficiency of regenerative braking	-
η_r	Energy efficiency of propulsion force	-
θ	Road grade	-
κ	Time step	-
λ	Regularization parameter	-
ρ_s	Spearman's rank correlation coefficient	-
ρ_τ	QR's check function	-
$\sigma(\cdot)$	Standard deviation	-
σ_{act}	Activation function	-
σ_{sig}	Sigmoid function	-
τ	Quantiles	-
ϕ	Random parameter vector	-
Θ	Parameters of the hidden layer	-
Σ	The expected number of time-series data samples	-
Ω	Regulation term	-

Bibliography

[1] I. P. on Climate Change, *Climate change 2021: The physical science basis*, Sixth Assessment Report, 2021. [Online]. Available: `https://www.ipcc.ch/report/ar6/wg1/` (cit. on p. 3).

[2] W. R. Institute, *Greenhouse gas emissions by gas*, 2020. [Online]. Available: `https://www.wri.org/data/greenhouse-gas-emissions-gas` (cit. on p. 3).

[3] U. N. E. Programme, *Emissions gap report 2021: The heat is on*, 2021. [Online]. Available: `https://www.unep.org/resources/emissions-gap-report-2021` (cit. on p. 3).

[4] G. J. Offer, "Automated vehicles and electrification of transport," *Energy Environ. Sci.*, vol. 8, no. 1, pp. 26–30, 2015 (cit. on p. 3).

[5] E. M. Bibra, E. Connelly, S. Dhir *et al.*, "Global EV outlook 2022: Securing supplies for an electric future," *Int. Energy Agency*, 2022 (cit. on p. 3).

[6] G. Crabtree, "The coming electric vehicle transformation," *Science*, vol. 366, no. 6464, pp. 422–424, 2019 (cit. on p. 3).

[7] D. Tan, "Transportation electrification: Challenges and opportunities," *IEEE Power Electron. Mag.*, vol. 3, no. 2, pp. 50–52, 2016 (cit. on p. 3).

[8] X. Hu, S. J. Moura, N. Murgovski, B. Egardt and D. Cao, "Integrated optimization of battery sizing, charging, and power management in plug-in hybrid electric vehicles," *IEEE Trans. Contr. Syst. Technol.*, vol. 24, no. 3, pp. 1036–1043, 2015 (cit. on p. 3).

[9] Y. Xu, S. Çolak, E. C. Kara, S. J. Moura and M. C. González, "Planning for electric vehicle needs by coupling charging profiles with urban mobility," *Nat. Energy*, vol. 3, no. 6, pp. 484–493, 2018 (cit. on p. 3).

[10] D. Baek, Y. Chen, A. Bocca *et al.*, "Battery-aware operation range estimation for terrestrial and aerial electric vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5471–5482, 2019 (cit. on p. 3).

[11] R. Basso, B. Kulcsár and I. Sanchez-Diaz, "Electric vehicle routing problem with machine learning for energy prediction," *Transp. Res. B*, vol. 145, pp. 24–55, 2021 (cit. on p. 3).

[12] Y. Zhang, X. Qu and L. Tong, "Optimal eco-driving control of autonomous and electric trucks in adaptation to highway topography: Energy minimization and battery life extension," *IEEE Trans. Transport. Electrif.*, vol. 8, no. 2, pp. 2149–2163, 2022 (cit. on p. 3).

[13] J. Zhang, T.-Q. Tang, Y. Yan and X. Qu, "Eco-driving control for connected and automated electric vehicles at signalized intersections with wireless charging," *Appl. Energy*, vol. 282, p. 116 215, 2021 (cit. on p. 3).

[14] M. Sabri, K. A. Danapalasingam and M. F. Rahmat, "A review on hybrid electric vehicles architecture and energy management strategies," *Renew. Sustain. Energy Rev.*, vol. 53, pp. 1433–1442, 2016 (cit. on p. 3).

[15] G Correa, P. Muñoz and C. Rodriguez, "A comparative energy and environmental analysis of a diesel, hybrid, hydrogen and electric urban bus," *Energy*, vol. 187, p. 115 906, 2019 (cit. on p. 3).

[16] H. Abdelaty, A. Al-Obaidi, M. Mohamed and H. E. Farag, "Machine learning prediction models for battery-electric bus energy consumption in transit," *Transp. Res. D*, vol. 96, p. 102 868, 2021 (cit. on p. 4).

[17] J. Zhang, Z. Wang, P. Liu and Z. Zhang, "Energy consumption analysis and prediction of electric vehicles based on real-world driving data," *Appl. Energy*, vol. 275, p. 115 408, 2020 (cit. on pp. 7, 34, 41).

[18] Chinese National Technical Committee of Auto Standardization, "Part 3: Communication protocol and data format," *Technical specifications of remote service and management system for electric vehicles*, 2016 (cit. on p. 7).

[19] D. Blatná, "Outliers in regression," *Trutnov*, vol. 30, pp. 1–6, 2006 (cit. on p. 9).

[20] C. Fiori, K. Ahn and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Appl. Energy*, vol. 168, pp. 257–268, 2016 (cit. on p. 13).

[21] Y. Zhang, T. Wik, J. Bergström, M. Pecht and C. Zou, "A machine learning-based framework for online prediction of battery ageing trajectory and lifetime using histogram data," *J. Power Sources*, vol. 526, p. 231 110, 2022 (cit. on p. 16).

[22] A. Lehman, N. O'Rourke, L. Hatcher and E. Stepanski, *JMP for basic univariate and multivariate statistics: methods for researchers and social scientists.* Sas Institute Inc., 2013 (cit. on p. 16).

[23] A. Katrutsa and V. Strijov, "Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria," *Expert Syst. Appl.*, vol. 76, pp. 1–11, 2017 (cit. on p. 16).

[24] R. J. Janse, T. Hoekstra, K. J. Jager *et al.*, "Conducting correlation analysis: Important limitations and pitfalls," *Clin. Kidney J.*, vol. 14, no. 11, pp. 2332–2337, 2021 (cit. on p. 16).

[25] C. Chatfield, "Calculating interval forecasts," *J. Bus. Econ. Stat.*, vol. 11, no. 2, pp. 121–135, 1993 (cit. on p. 18).

[26] S. Makridakis and M. Hibon, “Accuracy of forecasting: An empirical investigation,” *J. R. Stat. Soc. Ser. A (Gen.)*, vol. 142, no. 2, pp. 97–125, 1979 (cit. on p. 18).

[27] C. Williams and C. Rasmussen, “Gaussian processes for regression,” *Adv. Neural Inf. Process. Syst.*, vol. 8, 1995 (cit. on p. 19).

[28] R. Koenker and G. Bassett, “Regression quantiles,” *Econometrica*, vol. 46, no. 1, pp. 33–50, 1978, ISSN: 00129682, 14680262. (visited on 08/09/2023) (cit. on p. 20).

[29] W.-Y. Loh, “Classification and regression trees,” *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, vol. 1, no. 1, pp. 14–23, 2011 (cit. on p. 20).

[30] T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,” in *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, 2016, pp. 785–794 (cit. on p. 21).

[31] Q. Wang, Y. Ma, K. Zhao and Y. Tian, “A comprehensive survey of loss functions in machine learning,” *Ann. Data Sci.*, pp. 1–26, 2020 (cit. on p. 21).

[32] J. Friedman, T. Hastie and R. Tibshirani, “Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors),” *Ann. Stat.*, vol. 28, no. 2, pp. 337–407, 2000 (cit. on p. 22).

[33] C. Chen, “A finite smoothing algorithm for quantile regression,” *J. Comput. Graph. Statist.*, vol. 16, no. 1, pp. 136–164, 2007 (cit. on p. 23).

[34] N. Meinshausen and G. Ridgeway, “Quantile regression forests.,” *J. Mach. Learn. Res.*, vol. 7, no. 6, 2006 (cit. on p. 24).

[35] Y. LeCun, Y. Bengio and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015 (cit. on p. 26).

[36] D. E. Rumelhart, G. E. Hinton and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986 (cit. on p. 26).

[37] J. W. Taylor, “A quantile regression neural network approach to estimating the conditional density of multiperiod returns,” *J. Forecasting*, vol. 19, no. 4, pp. 299–311, 2000 (cit. on p. 27).

[38] Z. Li and D. Hoiem, “Learning without forgetting,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, 2017 (cit. on p. 28).