IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 8, 2020, accepted November 17, 2020, date of publication November 24, 2020,
date of current version December 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3039815

Machine Learning-Based Method for Remaining Range Prediction of Electric Vehicles

LIANG ZHAO[1,2,3], WEI YAO[4], YU WANG[1,2,3], AND JIE HU[1,2,3]

[1]Hubei Key Laboratory of Advanced Technology for Automotive Components, Wuhan University of Technology, Wuhan 430070, China
[2]Hubei Collaborative Innovation Center for Automotive Components Technology, Wuhan University of Technology, Wuhan 430070, China
[3]Hubei Research Center for New Energy and Intelligent Connected Vehicles, Wuhan University of Technology, Wuhan 430070, China
[4]Foreign Language School, Wuhan University of Technology, Wuhan 430070, China

Corresponding author: Wei Yao (yaoweivivian@126.com)

This work was supported by the Liuzhou Science and Technology Plan Project (Key Research and Development Plan) under Grant 2018B0301b003.

ABSTRACT Limited driving range is one of the major obstacles to the widespread application of electric vehicles (EVs). Accurately predicting the remaining driving range can effectively reduce the range anxiety of drivers. In this paper, a blended machine learning model was proposed to predict the remaining driving range of EVs based on real-world historical driving data. The blended model fuses two advanced machine learning algorithms of Extreme Gradient Boosting Regression Tree (XGBoost) and Light Gradient Boosting Regression Tree (LightGBM). The proposed model was trained to ''learn'' the relationship between the driving distance and the proposed features such as cumulative output energy of the motor and the battery, different driving patterns, and temperature of the battery). In addition, an ''anchor (baseline) based'' strategy was proposed and was seen to be able to effectively eliminate the unbalance distribution of dataset. The results of experiments suggest that our proposed anchor-based blended model has better performances with a smaller prediction error range of [-0.8, 0.8] as compared with previous methods.

INDEX TERMS Electric vehicle, remaining driving range estimation, machine learning, data mining.

I. INTRODUCTION

Electric vehicles (EVs) are regarded as a promising means of transportation to reduce air pollution and fossil fuel consumptions [1]–[4]. The major bottleneck for further deployment of EVs is related to the power batteries. Lithium-ion batteries are being used predominantly as energy storage devices in EVs as a result of their high energy density, long cycle life, and wide operating temperature. However, there are also issues that hinder the penetration of EVs worldwide; these include, for instance, battery degradation [5], [6], cell inconsistency [7], [8], and thermal runaway during overcharge [9]–[12]. The lithium-ion battery inevitably suffers from capacity degradation during their service lifetime since the batteries always have side reactions. Generally speaking, a battery would be regarded to have reached its end of life (EOL) [5] with a capacity fade of 20% or an internal resistance increase of 100%. The battery state of health (SOH) reflects the capacity of the battery to some extent. However, the estimation of SOH cannot be made very precise due to the complex internal reactions of batteries. Cell inconsistency (i.e., cell to cell variation), which usually derives from the manufacturing process and will get even worse during battery operation, is regarded as the main influencing factor for SOH [7]. There are inevitably deviations in active materials, formation of solid electrolyte interphase (SEI) film, and electrode thickness in the manufacturing process of battery cells. The uneven load current and thermal distribution would further induce cell inconsistency or even cause battery thermal runaway accidents [7], [13]. The level of cell inconsistency can be reflected by the variations in the output energy, terminal voltage, temperature, SOC, and etc. [7]. Therefore, these variables may indicate the state of the battery to some extent.

Due to battery degradation and the difficulty in the estimation of SOH, accurate prediction of remaining driving range of EVs is usually quite challenging. This in turn causes the ''range anxiety'' of drivers, which is defined as the psychological anxiety that drivers suffer from worrying about whether their EV can arrive at the destination before exhausting the battery [14], [15].

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

According to data collected by the National Big Data Alliance of New Energy Vehicles (NDANEV) [16], most drivers recharge their EVs before the state of charge (SOC) drops below 25%, which due to the range anxiety [14] and other reasons (e.g. the reduction of power capability [18], and quickly speed up of battery aging [19]). Among these factors, range anxiety is one of the major issues for EVs, due to their limited all-electric-range (AER) and long battery recharge times [20]. Accurately predicting the remaining driving range of EVs can offer the drivers more precise information about their remaining driving range and reduce the range anxiety [21].

Predicting the remaining driving range is usually a combination of predicting the energy consumption rate and the remaining capacity of the battery. The latter is typically represented by the SOC. Nevertheless, battery degradation makes it difficult to create a direct linkage between remaining energy and SOC [17], [22]. Though the Coulomb counting is developed widespread for SOC estimation in commercial battery management systems (BMSs), this method is sensitive to noise, parameter uncertainties, and deviation in initialization [23]. Therefore, the estimation accuracy of SOC is not only affected by battery degradation and also affected by the estimation error of BMSs. Therefore, some novel features (e.g. cumulative output energy of the motor and the battery) were proposed in this paper, which combined with SOC can effectively improve the prediction accuracy of the remaining driving range.

There are also many factors that would affect the energy consumption rate, such as speed, driving patterns, braking frequency, acceleration frequency, battery temperature, and auxiliary loads (air conditioning, ventilation, lights, the horn, etc.), among others. Existing studies in the literature mainly focused on exploring the effect of external factors on the energy consumption rate. The authors of Ref. [24], [25] analyzed the impact of external temperature and auxiliary loads (e.g. light, air condition, etc.) on energy consumption rate. In Ref. [26], the influence of the road gradient on energy consumption rate was considered. However, using the external factors to represent the energy consumption rate usually may not be precise enough as it is impossible to comprehensively consider all the external factors. In addition, besides the many external factors, driver behaviors such as driving patterns, speed and acceleration may also affect the energy consumption rate. Indeed, the impact of vehicle speed and acceleration on the energy consumption rate was analyzed in Ref [21]. However, future speed profile is difficult to predict.

Machine learning-based approaches have been employed for the remaining driving range prediction of EVs [27], [28]. In existing studies, the adopted algorithms seem to be relatively traditional; examples include multiple linear regression (MLR) [27], [29], neural network [20], [30], gradient boost decision tree (GBDT) [31]. In recent years, data scientists have proposed a variety of novel machine learning algorithms such as XGBoost [32] and LightGBM [33], which have been proven to have better performance than traditional methods in a number of application fields [34]–[37].

However, to the best of the authors' knowledge, there is no previous study in which the XGBoost or LightGBM has been used to predict the remaining driving range of EVs. Our experiments, as detailed in a later Section (Section IV.A), proved that XGBoost and LightGBM indeed have better performance in remaining range prediction than other traditional methods. Furthermore, we have proposed a blended model of XGBoost and LightGBM to further improve the prediction accuracy of the remaining driving range.

The unbalance distribution of training data is one of biggest challenges in machine learning studies. Generally speaking, if the training data and testing data follow different distributions, the prediction accuracy of machine learning models on testing data will be negatively affected. However, previous methods not mentioned this problem, they just simply use the driving distance as the regression targets for machine learning models. In this paper, we proposed an anchor (baseline) based strategy, which can effectively solve the unbalance distribution of training data.

In the present work, we directly calculate the cumulative output energy of the motor (*COEM*) to represent the energy that are consumed for driving the vehicle. Combining *COEM* and *COEB* as the features can avoid estimating the complex non-driving-energy. The levels of energy consumption are represented by the driving patterns, which are obtained through the K-means cluster algorithm. In brief, we proposed in this work a two-stage framework to predict the remaining driving range of EVs. The first stage aims to train the blended machine learning model by using features and the label (calculated through the odometer). The second stage uses the proposed feature prediction model to predict future values of these features based on historical values. Then, the predicted features are used as inputs of the trained blended model to get the remaining driving range of EVs.

Our contributions can be summarized into the following three aspects: 1) We proposed a two stage framework for remaining driving range prediction of EV, which deeply fuses two advanced machine learning algorithms of XGBoost and LightGBM and is confirmed to be able to improve the prediction accuracy of the remaining driving range; 2) We adopted the ''anchor-based'' strategy in the remaining driving range prediction, which can effectively eliminate the unbalance issue of the training label. The anchor-based model is more robust than the anchor-free model on the different testing data. 3) We proposed various driving features such as *COEB*, *COEM*, *BR, RDP,* etc. (see Table 2), which distinguished with previous methods and can effectively improve the accuracy of remaining driving range prediction.

The remaining sections of this paper is organized as follows. A literature survey on related works are presented in section II. The methodology is described in Section III. In, the comparison experiments are discussed in section IV along with comparisons being made with existing works to

highlight the novelty of our method. The main conclusions are summarized in Section V.

II. RELATED WORKS

There are two solutions to relieve range anxiety, 1) predicting the energy consumption rate, and 2) predicting the remaining driving range directly.

A. PREDICTING THE ENERGY CONSUMPTION RATE OF EV

Some researchers focused on accurately predicting the energy consumption rate of EV to reduce the range anxiety. Halmeaho *et al.* [38] established an electric bus simulation model in which a motor efficiency curve model and a resistance model were utilized to predict the energy consumption rate. Fiori *et al.* [39] considered the regenerative braking energy efficiency at different vehicle speeds to predict energy consumption with an average error of 5.9%. De Cauwer *et al.* [27] adopted a neural network to predict the future speed profile based on driving data of 30 electric vehicles and then used multiple linear regression (MLR) model to predict the energy consumption rate of EVs through predicted speed and other factors. Predicting the energy consumption rate can alleviate range anxiety to some extent, but it cannot offer precise information about the remaining driving range to the driver.

B. PREDICTING THE REMAINING DRIVING RANGE DIRECTLY

The approaches to predict the remaining driving range in the literature can be classified into two categories: 1) simulation based method and 2) data-driven based method. Simulation-based approaches [40], [41] can achieve high prediction accuracy but require high fidelity of the simulation model. In addition, calibration according to a specific vehicle are needed. Data-driven based approaches are independent on physical models. However, they require high-quality data preprocessing and feature engineering, as the raw datasets may include a lot of irrelevant data.

1) SIMULATION-BASED METHODS

Genikomsakis and Mitrentsis [40] established a simulation model for batteries, motors, and vehicle drivetrains to estimate the remaining range of EVs. Hayes *et al.* [41] proposed a simplified powertrain model of EV to predict the remaining driving range. Simulation-based methods usually need to establish a simulation model through the specific vehicle and detailed internal vehicle parameters; so it is hard to generalize. As a result, some researchers have turned their attention to the data-driven method (generally, it is a regression model) to predict the remaining driving range.

2) DATA-DRIVEN BASED METHODS

Data-driven based methods have the merits of simplicity as they do not need a physical model of vehicle. In the data-driven method, the constant updating of driving data facilitates real-time prediction of the remaining driving range.

TABLE 1. The raw data Items.

Column names of raw data	Description
Speed	Longitudinal speed
V_{motor}	Motor voltage
I_{motor}	Motor current
$V_{battery}$	Battery pack voltage
$I_{battery}$	Battery pack current
SOC	State of charge
Temp_max	Maximum cell temperature
Temp_min	Minimum cell temperature
Mileage	Odometer value
t	Timestamp

Data-driven methods usually involve using a regression model fitted to historical driving data to predict the remaining driving range under the assumption of constant values of driving parameters. In Ref. [28], a growing hierarchical self-organizing map (GHSOM) was used to perform clustering to obtain the relationship between vehicle speed and energy efficiency; and then the accelerated battery aging experiments were performed to obtain the battery capacity profile. The remaining range is estimated by the SOC and energy efficiency. Bi *et al.* [42] found that the relationship between speed and **d**riving **d**istance **p**er **S**OC (*DDPS*) follows a Weibull distribution. Therefore, they used the Weibull distribution to predict the *DDPS* through vehicle speed. SOC and *DDPS* were then used to calculate the remaining driving range. Some other authors [20], [25], [26], [31] adopted traditional machine learning methods such as multiple linear regression model (MLR), radial basis function neural network (RBF NN), and gradient boost decision tree (GBRT) to predict the remaining range.

III. REMAINING RANGE PREDICTION FRAMEWORK

The remaining range prediction framework we proposed in this work consists of two stages. The first stage aims to train the machine learning models to ''learn'' the relationship between the features and the label (i.e. driving distance calculated through the odometer value). And in the second stage, the predicted features will be taken as the inputs of the trained blended model to predict the remaining driving range.

A. FIRST STAGE: TRAINING THE MACHINE LEARNING MODEL

In this work, the first stage is mainly about algorithm training. The raw data includes battery parameters, motor parameters, and driving logs. These data items are used to construct features related to the driving distance, as demonstrated in Fig 1. Note, all the terms mentioned in Fig 1 are described in Section III.E. The principle of these machine learning algorithms will be presented in Section III.C.

B. SECOND STAGE: PREDICTING THE REMAINING DRIVING RANGE

The flow chart of the second stage is shown in Fig 2. The raw data items (shown in Tables I) are obtained from T-Box, and

TABLE 2. The features.

Feature name	Description
----	$\mathbf{t_0}$ denotes the initial timestamp of a trip. **t** denotes the current timestamp.
Temp_max(t) [℃]	Maximum temperature of cell.
Temp_min(t) [℃]	Minimum temperature of cell.
Temp_diff(t) [℃]	Difference of Temp max and Temp min.
COEM (t_0, t) [J]	**C**umulative **O**utput **E**nergy of the **M**otor from t_0 to t.
COEB(t_0, t) [J]	**C**umulative **O**utput **E**nergy of the **B**attery from t_0 to t.
----	The relationship between the Used_SOC and COEB can reflect the capacity degradation of batteries to some extent.
BR(t_0, t) [%]	**B**raking **R**atio from t_0 to t.
SR(t_0, t) [%]	**S**topping **R**atio from t_0 to t.
AR(t_0, t) [%]	**A**cceleration **R**atio from t_0 to t.
Used_SOC(t_0, t) [%]	The drops value of SOC from t_0 to t.
Driving time(t_0, t) [s]	The difference of current timestamp and start timestamp.
RDP_1(t_0, t), RDP_2(t_0, t), RDP_3(t_0, t), RDP_4(t_0, t) [%]	Cumulative **R**atio of different **D**riving **P**atterns from t_0 to t

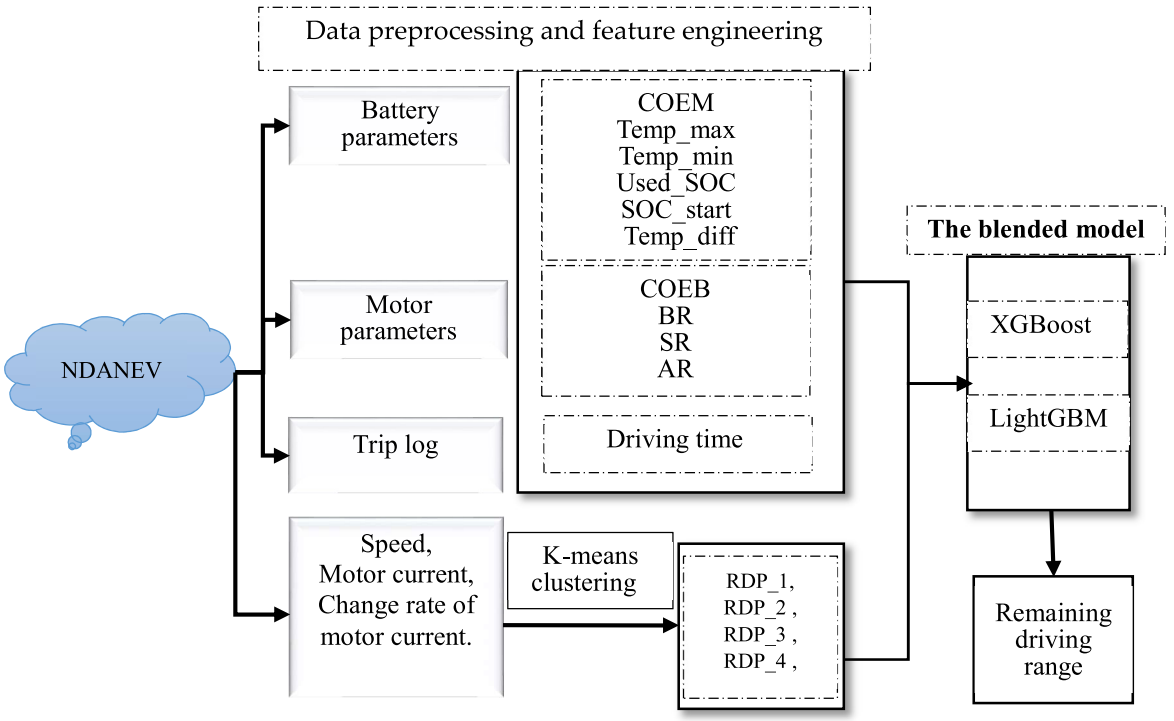

FIGURE 1. First stage: training the machine learning algorithms. NDANEV is an EV cloud. Above mentioned terms are descripted in Table 2 and Section III.E.

then the features of current timestamp (as shown in Table 2) are calculated through formula (6) - (12) (described in section III.E).

In Fig 2, the $COEM(t_0, t)$ can be calculated by $V_{\text{motor}}, I_{\text{motor}}$ and timestamp through formula (6) and (7), in which t denotes current timestamp and t_0 denotes the initial timestamp of a trip.

In Fig 2, the $COEM(t_0, t)$ can be calculated by $V_{\text{motor}}, I_{\text{motor}}$ and timestamp through formula (6) and (7), in which t denotes current timestamp and t_0 denotes the initial timestamp of a trip. The feature $Used\_SOC(t_0, t)$ can be calculate as $SOC(t_0) - SOC(t)$. Then, $COEM(t_0, t)$ and $Used\_SOC(t_0, t)$ were respectively appended to one dimension arrays as: $COEM\_list(t)=\{COEM(t_0, t_1), COEM(t_0, t_2), \ldots COEM(t_0,t)\}$ and $Used\_SOC\_list=\{Used\_SOC(t_0, t_1), \ldots, Used\_SOC(t_0,t)\}$. After a short interval, if the length of the array exceeds a predefined threshold, the relationship between $Used\_SOC\_list(t)$ and $COEM\_list(t)$ can be fitted to get the feature prediction model of $COEM$ (denoted as M). Then, $Used\_SOC(t, T)$ can be used as the input of M to get the $COEM(t, T)$ (denoted as f_1). Here T denotes the timestamp when SOC equals zero. Finally, we take the f_1 along with other predicted features (see section III.G) as the inputs of the blended model to get the remaining driving range.

C. PRINCIPLE OF XGBOOST, LIGHTGBM AND THE BLENDED MODEL

The principles of the Gradient Boost Regression Tree (GBRT), XGBoost, and LightGBM algorithms are described in detail in [46], [32], and [33], respectively.

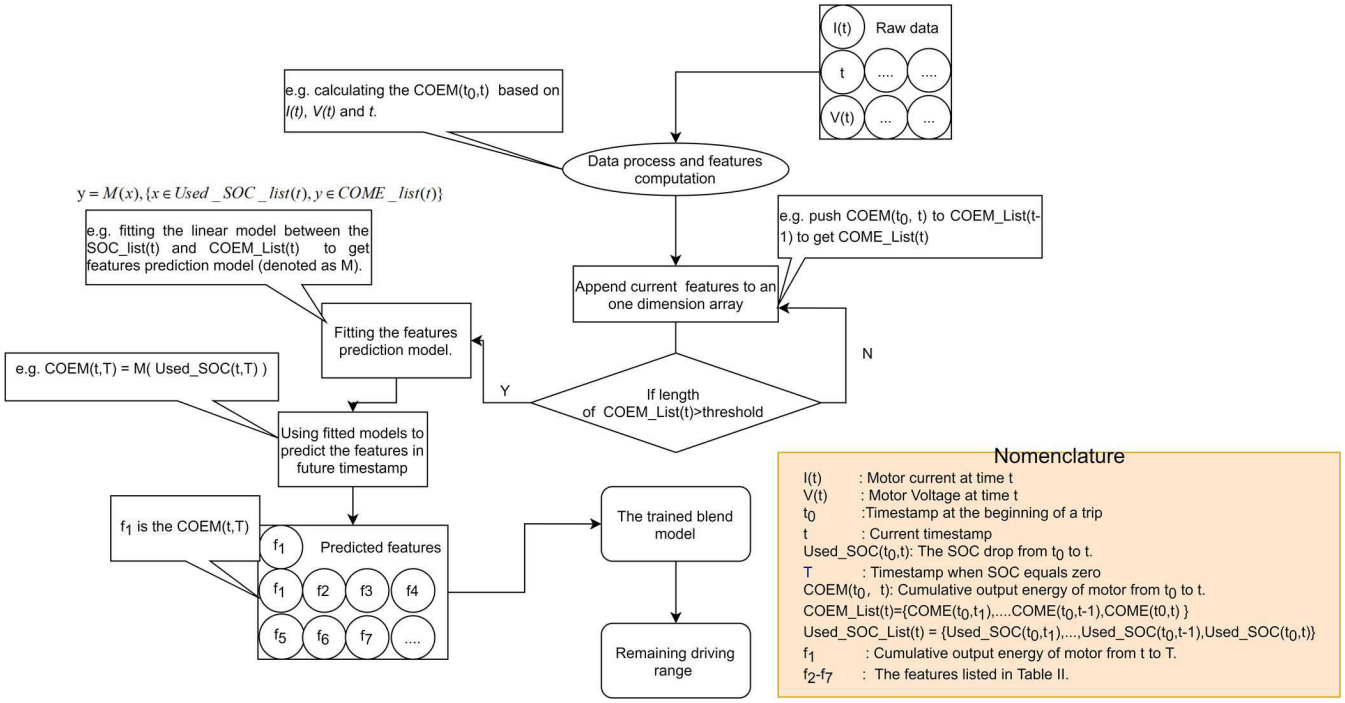

FIGURE 2. Second stage: the predicted features will be used as the inputs of the blended model to predict the real remaining driving range.

These three ensemble learning algorithms use the decision tree as the base learner and adopt boosting ideas to gather many base learners to achieve high prediction accuracy.

XGBoost is a boosting machine learning algorithm. Its central principle is that it combines many weak base learners in an ensemble to boost its performance. The base learner usually is a regression tree. To train the model, an objective function is needed to measure how well the model fits the training data, as shown in formula (1). The objective function consists of two parts:

$$obj\,(\theta) = L\,(\theta) + \Omega(\theta) \tag{1}$$

a training loss term $L(\theta)$ and a regularization term $\Omega(\theta)$, θ denotes the parameters that the regression model has learned. XGBoost approximates to represent the training loss $L(^*)$ by using a second-order Taylor expansion, as shown in formula (2).

$$L\,(\theta) = \sum_{i=1}^{n} [L\left(y_i, \hat{y}_i^{(t-1)}\right) + g_i f_t\,(x_i) + \frac{1}{2} h_i f_t^2(x_i)] \tag{2}$$

where y_i denotes true value, $\hat{y}_i^{t-1}$ denotes the prediction of the i^{th} instance at the $(t\text{-}1)^{\text{th}}$ iteration. g_i and h_i are the first and second derivative of $l(y_i, \hat{y}_i^{(t-1)})$ to $\hat{y}_i^{(t-1)}$ respectively. The training process proceeds in an additive manner, as shown in formula (3).

$$\hat{y}_i^{\,t} = \hat{y}_i^{(t-1)} + f_t(x_i) \tag{3}$$

Here, $\hat{y}_i^{t}$ denotes the prediction of the i^{th} instance at the t^{th} iteration. We define the tree $f_t(x)$ as shown in formula (4):

$$f_t\,(x) = \omega_{q(x)} \tag{4}$$

where $q(x)$ is a function that assigns each data point to a corresponding leaf, ω is the vector of scores on leaves. The regularization term is expressed as shown in (5).

$$\Omega\,(\theta) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{5}$$

where ω_j is the vector of scores on leaves j, and T is the number of leaves. γ and λ are constant values. Learning a tree structure is much harder than a traditional optimization problem that simply requires determining the gradient. XGBoost uses an additive strategy, fixing what has been learned and adding one new tree at a time, as shown in formula (3), in which f_t is added to minimize the objective function.

An optimal tree structure needs to be chosen at each step, and XGBoost uses the objective function expressed by (1) to optimize the objective. The main principle of LightGBM is the same as that of XGBoost.

Furthermore, we proposed a blended model that deeply fuses XGBoost and LightGBM, which like a neural network as shown in Fig 3.

XGBoost_1 and LightGBM_1 share the same features and then get the different output: *output_1* and *output_2*. The *output_1* and *output_2* correspond to the same label *T and* also are the input of XGBoost_2. The blended model refines the remaining driving range regression in the second layer: XGBoost_2.

D. DATA COLLECTION

The raw data used in this study is collected from NDANEV [16]. New energy vehicles usually are equipped

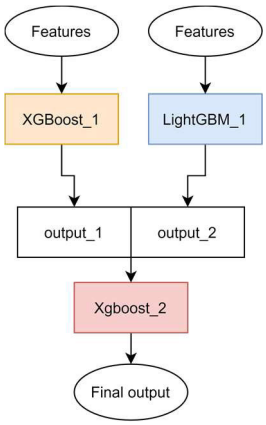

FIGURE 3. The blended model scheme, the features are listed in Section III.E.

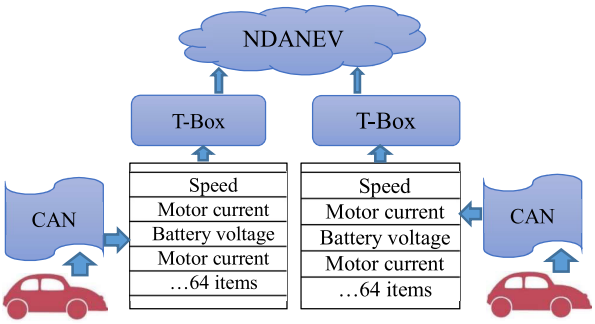

FIGURE 4. Schematic diagram of the NDANEV platform. Real-world driving data of EVs are collected from each vehicle's CAN bus.

with T-BOX which collects the driving data from Controller Area Network (CAN) and uploads these data to NDANEV every 10 seconds, as shown in Fig. 4. The raw data items are described in Table 1.

Our raw dataset consists of more than two thousand trips (covering more than 600,000 km) of five battery-powered electric vehicles with the same model in Beijing, China. Because of a confidentiality agreement, the authors of this study are not allowed to disclose the specific information of the electric vehicles used.

Up to now, NDANEV has connected 1.358 million new energy vehicles (NEVs). NDANEV uses the distributed file system Hadoop [43]–[45] as the basic system framework.

E. FEATURE ENGINEERING

The raw dataset contains a lot of non-driving data; so the first step is data clean (deleting data related to charging and parking). The preprocessing pipeline is described as follows.

Step 1: We sorted the data according to the timestamp.

Step 2: We dealt with the outlier data. Specifically, we adjusted the outliers and filled in the missing values according to the nearest point of these incorrect data.

Step 3: We up-sampled the data from the sampling frequency of 0.1 Hz to 1 Hz through linear interpolation.

Step 4: We constructed the interval values for three variables, timestamp, SOC, and values of the odometer (e.g the interval value for timestamp, *timestamp(i) – timestamp(i-1)*, *i* denotes the current sample index). Then, we used these three interval values to slice the raw dataset into many individual trips.

Step 5: Each of the driving features were constructed in every individual trip.

Step 6: We manually added some noise in every features values for preventing overfitting. For instance, feature a will became $a \pm a* \ noise\_ratio$, the *noise_ratio* is in the range of [0, 1], which denotes the noise level.

The features, as shown in Table 2, are constructed in every timestamp. The remaining energy of an EVs is usually represented by SOC. However, the capacity of the battery will degrade as the use time increases. It is well known that the remaining energy of the old battery and the new battery is different under the same SOC value. Therefore, in this study, to express the energy used for driving more accurately, we calculate $COEM(t_0, t)$ through motor voltage, motor current, and timestamp, as shown in formula (6) and (7).

$$COEM\ (t_0, t) = COEM\ (t_0, t-1) + \int_{t-1}^{t} P_{motor}(x)dx \quad (6)$$

$$P_{motor}\ (x) = V_{motor}\ (x) \times I_{motor}\ (x)\,, \quad x \in [t-1, t) \quad (7)$$

where $V_{\text{motor}}(x)$ and $I_{\text{motor}}(x)$ are the motor voltage and the motor current, respectively. t denotes current timestamp, t_0 denotes the initial timestamp of a trip, $COEM(t_0, t)$ denotes the cumulative output energy from t_0 to t, P_{motor} denotes the power of motor. In the same way, we can calculate $COEB(t_0, t)$ by replacing $I_{\text{motor}}(x)$ and $V_{\text{motor}}(x)$ with $I_{\text{battery}}(x)$ and $V_{\text{battery}}(x)$, respectively in formula (7). The driving time can be calculated through formula (8).

$$T_d\ (t) = t - t_0 \quad (8)$$

The inconsistency of the battery reflects the state of the battery to some extent. In this paper, the difference between the maximum and the minimum temperature of cells (denoted as *Temp_diff*) is used to indicate the cell inconsistency, and it is calculated through formula (9)

$$Temp\_diff\,(t) = Temp\_max(t) - Temp\_min(t) \quad (9)$$

The training label is the driving distance, which is calculated through formula (10).

$$driving\_distance(t) = mileage(t) - mileage(t_0) \quad (10)$$

mileage(t) denotes the current odometer value and $mileage(t_0)$ denotes the odometer value at the initial moment of the driving trip.

We use the sign of the motor current to distinguish the vehicle state information such as acceleration, stopping, and braking. If the sign of motor current is positive, the vehicle is at acceleration or at a constant speed (denoted as [1, 0, 0]). If the motor current equals zero, the vehicle is at a stop state (denoted as [0, 1, 0]). If the sign of motor current is negative, then the vehicle is at regenerative braking (expressed as

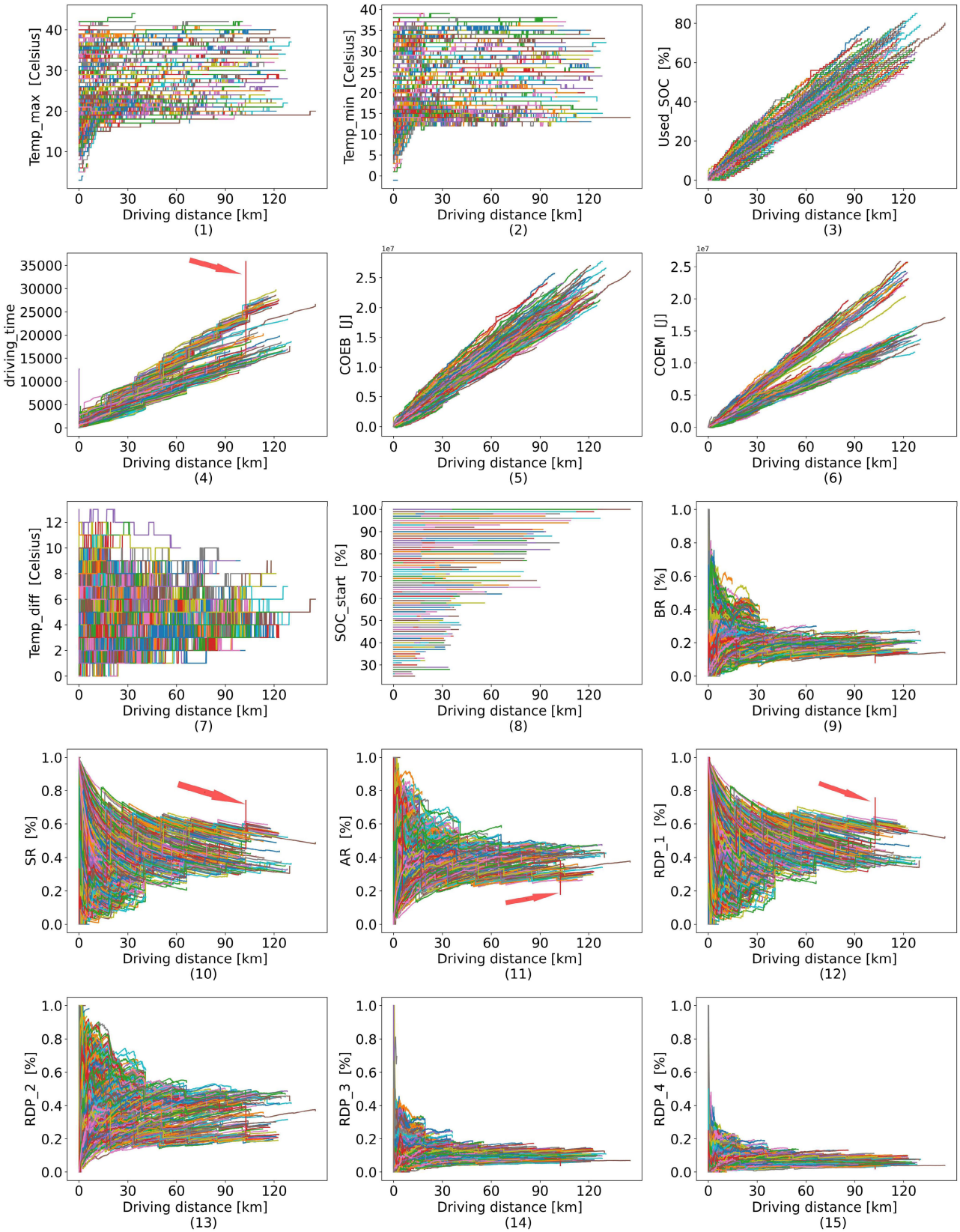

FIGURE 5. Relationships between features and driving distance. Every subplot contains more than 2,000 trips and the lines with different colors represent different trips. The red arrows pointed some positions that exhibit abnormal fluctuations. The SOC_start denotes the initial SOC of a trip. In (5), 1e7 means every ordinate value should multiply 1e7(10^ 7).

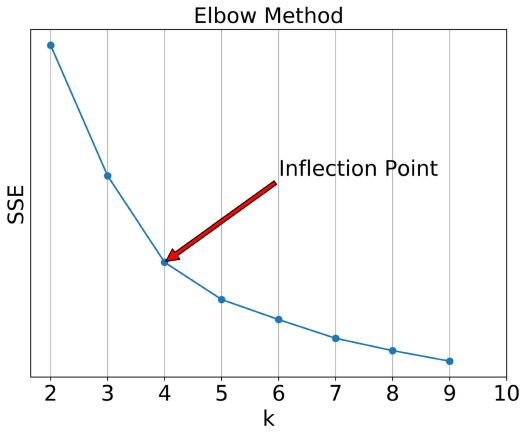

FIGURE 6. "Elbow" method for choosing the number of cluster centers.

[0, 0, 1]). We should change the transient state (braking, stopping, acceleration at time t) to the cumulative ratio: braking ratio (BR), stopping ratio (SR), and acceleration ratio (AR).

For example, there are three consecutive transient vehicle states:

$$t_1[0, 0, 1], t_2[0, 1, 0], t_3[0, 0, 1]$$

Firstly, we calculate the cumulative state t_{1-3} [0, 1, 2] in an additive manner. Then, the $AR(t_1, t_3)$, $SR(t_1, t_3)$, and $BR(t_1, t_3)$ from moment t_1 to t_3 are 0/(0+1+2), 1/(0+1+2) and 2/(0+1+2)respectively. Therefore, we summarized the calculation as the formula (11).

$$RBE_i\,(t_0, t) = \frac{\sum_{t_x=t_0}^{t} rbe_i^{t_x}}{\sum_{i=1}^{3}(\sum_{t_x=t_0}^{t} rbe_i^{t_x})} \tag{11}$$

Here, $rbe_i^{t_x}$ denotes the i^{th} transient state (including three states: braking, stopping, and acceleration) at time t_{x}, $RBE_{\text{i}}(t_0, t)$ denotes the cumulative ratio of the i^{th} states from t_0 to t.

We adopt the K-means cluster algorithm to obtain these four types of driving patterns (i.e. dp_1 to dp_4) through clustering three variables: speed, motor current, and change rate of motor current, as described in Section III.F. dp_1 to dp_4 are defined as: [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1], respectively.

In the same way to AR, SR, and BR, the cumulative **r**atios of four **d**riving **p**atterns ($RDP\_1, \ldots, RDP\_4$) from t_0 to t is calculated through formula (12).

$$RDP_i\,(t_0, t) = \frac{\sum_{t_x=t_0}^{t} dp_i^{t_x}}{\sum_{i=1}^{4}(\sum_{t_x=t_0}^{t} dp_i^{t_x})} \tag{12}$$

Here, $dp_{\text{i}}^{\text{t}_{\text{x}}}$ denotes the driving pattern i at time t_{x}.

We exhibit the relationship between these features and the driving distance by using more than 2,000 trips, as shown in Fig. 5.

With increasing driving distances, the maximum and minimum temperatures of cells exhibit a characteristic of rising-and-plateauing, as shown in Fig. 5 (1) and (2). $Used\_SOC$, $COEB$ and $COEM$ are approximately linear with driving distance, as shown in Fig. 5 (3), (5), and (6). The relationships between driving distance and BR, SR and AR are respectively shown in Fig. 5. (9), (10) and (11). RDP_1 to RDP_4 are shown in Fig 5 (12)-(15).

F. DRIVING PATTERNS

In this section, we present a driving pattern recognition model that uses the k-means clustering algorithm to identify different driving patterns. The driving patterns represent the different levels of energy consumption rate.

Vehicle speed significantly affects energy consumption rate. Also, the motor current reflects driving mode information, such as the strength and frequency of the regenerative braking to some extent. The change rate of the motor current reflects the states of driving such as acceleration and braking. In this study, vehicle speed, motor current, and the change rate of motor current are clustered through K-means algorithm to obtain several driving patterns, according to the following steps.

Step 1: The elbow method is used to determine the number of clusters to be 4, as shown in Fig 6. SSE represents how closely the distance between the sample points and their cluster center, as shown in formula (13).

$$SSE = \sum_{j=1}^{k}(\sum_{p\in C_i} ||p_{ij} - \mu_i||^2) \tag{13}$$

where p_{ij} denotes the j^{th} sample point in i^{th} cluster. μ_i denotes the center of i^{th} cluster.

If the number of cluster centers (denoted as k) was set to be 4, there would be an "inflection". Further increase of the value of k led to only relatively small drop of SSE. So, the x-coordinate value of inflection point is the optimal number of cluster centers since more cluster centers will degrade its performance.

Step 2: The k-means clustering algorithm is used to cluster the vehicle speed, motor current, and change rate of motor current according to the number of clusters determined by the elbow method.

These four driving patterns can be described as normal driving, conservative mode, sport mode, and extreme mode. The K-means cluster algorithms can distinguish the different driving patterns but cannot give the corresponding relationship between the four driving patterns and the four modes assumed by the authors.

G. FEATURES PREDICTION MODEL

We have established the blended model that has "learned" the relationship between the driving distance and various driving features, as described in previous sections. However, the future values of these features are unknown, which are required as the input of the trained blended model. Therefore, we assume that the SOC drops from the current value to zero, and then predict the future values of these features based on the historical data. We found that $COEB(t_0, t)$ and $COEM(t_0, t)$ are approximately linearly

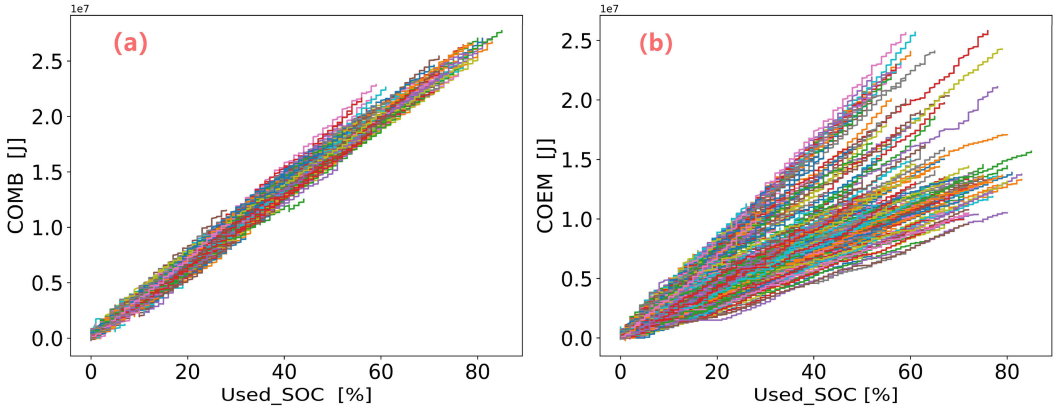

FIGURE 7. Relationship between the Used_SOC and COME as well as COMB. Every sub-plot contains more than 2000 trips. Different colors represent different trips.

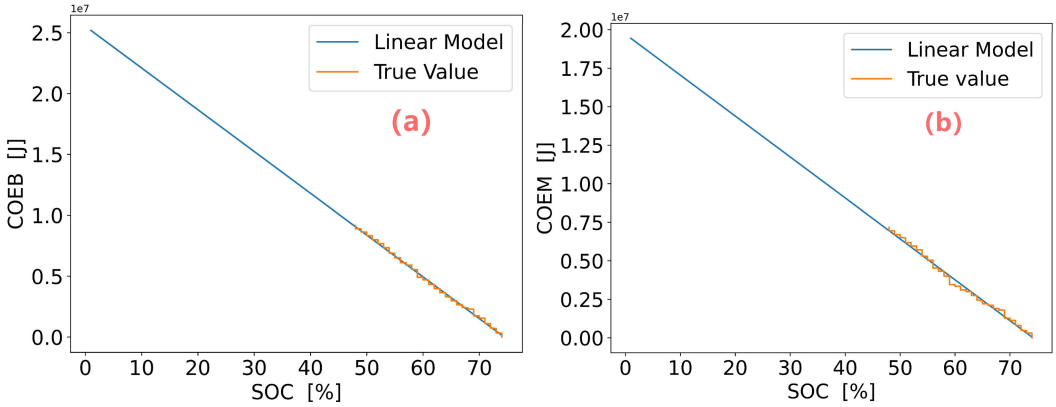

FIGURE 8. Using a linear regression model to predict COEB and COEM, (a) using a linear model to fit the true value and then predict the value of COEB from t to T, t denotes the current timestamp and T denotes the timestamp when SOC equals zero. (b) use a linear model to predict COEM.

related to $Used\_SOC(t_0, t)$. However, as shown in Fig. 7 (b), the slopes of *COEM* to *Used_SOC* are different in different trips as the values of *BR*, *SR*, *AR*, $RDP\_1, \ldots, RDP\_4$, and other factors vary among different trips. We used linear regression models to predict *COME(t, T)* and *COMB(t, T)* based on *Used_SOC(t, T)*.

For instance, we randomly selected a trip, as show in Fig 8. The initial *SOC* of this trip is 74%; as we assume *SOC* drops to zeros, *Used_SOC* equals to 74%. The *COEM(t,T)* and *COEB(t,T)* can be predicted by *Used_SOC(t,T)* through the fitted linear model, as shown in Fig. 8. As SOC decreases, the *COEB* and *COEM* will increase linearly. A linear regression model is adopted to constantly regress the relationship between the *COEB* and SoC, which can indirectly reflect the information about the battery capacity through the slop of the linear model. Therefore, the *COEB* and *COEM* can compensate for SOC to solving battery degradation, which is the reason why adding the *COEM* and *COEB* to the machine learning model can improve the prediction accuracy.

We found that the *Temp_max* and the *Temp_min* tend to rise first and then reach constant values with the increase of the driving distance, as shown in Fig. 5 (1) and (2). Therefore, we assume that the values of *Temp_max(t,T)* and *Temp_min(t,T)* remain within $\pm 2^\circ$ of current values $Temp\_max(t_0, t)$ and $Temp\_min(t_0, t)$, respectively. We also observed that *BR*, *SR*, and *AR* tend to exhibit large fluctuations initially and then become more stable as the driving distance increases, as shown in Fig. 5 (9), (10), and (11).

BR values are constantly updating based on the instantaneous motor current. At the initial phase of a trip, the future value of *BR* is unknown and is also very difficult to predict, mostly because of the highly unpredictable human behavior. Therefore, we are forced to assume that the current *BR* value represent the future profile. However, at the beginning of a trip, the *BR* is very unstable, which may cause obvious prediction error. Therefore, a threshold is needed to filter the unstable values of *BR*, as shown in Fig 9.

When the driving distance of the current trip less than the threshold (which equals to 10 km here), the algorithm will take the features from the end phase of the previous trip as current features. Most trips seem to be short-distance trips, i.e. the driving distance is less than 30 km. The *BR* values are roughly stable when driving distance were over 10 km in

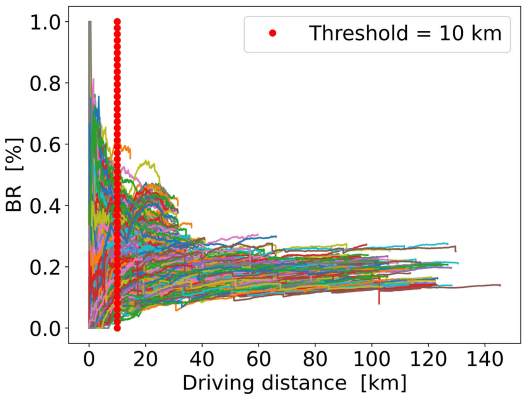

FIGURE 9. The strategy for choosing BR values. There are more than 2000 trips in this figure, the lines with difficult colors represent the different trips

their trips. If a large threshold is chosen, it may not suit for short-distance trips. On the other hand, if a small threshold is chosen, the initial phase of BR will become more unstable. Therefore, we compromise the threshold as 10 km, as shown in Fig.9. If the driving distance of the current trip is beyond the threshold, the proposed assumption that the current values represent the future profile will be adopted.

Though this strategy only gets coarse BR values, the BR ranks 6th in the ranking of features importance as shown in Fig 12, so the variations in BR can only have limited influences on the prediction of remaining driving range. To the best knowledge of the authors, there is no previous work that achieves accurate prediction of future BR at the initial phase of a trip. At the initial moment of a trip, we are unknown to the behavior of the driver, so we can only assume that the behavior of the driver on this trip will be similar to the previous trips.

The *RDP_1* to *RDP_4* are similar to the *BR*, *SR,* and *AR* in changing patterns, as shown in Fig 5 (13), (14), (15), (16).

The driving time can be predicted using a multiple linear regression model of the following formula:

$$Driving\ time = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_3 x_3 \qquad (14)$$

In formula (14), x_1, x_2, and x_3 denote value of $Used\_SOC(t_0,t)$, $BR(t_0, t)$ and $SR(t_0, t)$, respectively. β denotes coefficient values that gained from fitting this multiple linear regression model.

In real-world applications, the remaining driving range can be defined as the driving distance that the electric vehicles be driven from the current SOC value dropping to zero. However, according to statistics results on the dataset collected from NDANEV [15], the drivers often did not exhaust the battery, so the odometer only recorded the driving segment from the initial of a trip to the end of a trip rather than a full discharge process of EVs. Thus, the real remaining driving range of vehicles is unknown as the battery did not exhaust. Therefore, we should adjust the definition of remaining driving range to the driving distance that the electric vehicles can be driven from the current *SOC* value dropping to the *SOC(T)*, *T* is the timestamp when a trip reaches its end.

TABLE 3. Parameters values used in the machine learning algorithms. Default values are not listed.

GBRT Parameters	XGBoost Parameters	LightGBM Parameters
N_estimators:5500 Learning_rate:0.05 Max_depth:3 Max_features:0.5 Min_sample_leaf:1 Min_sample_split:2 Loss: Huber Subsample:0.5 n_iter_change:200	Num_boost_rounds:46538 Eta:0.05 Max_depth:3 Colsample_bytree:0.4603 Lambda:0.8571 Alpha:0.4640 Gamma:0.0468 Subsample:0.5213 Tree_method: exact Min_child_weight:1.7817 Early_stopping_rounds: 1000	Num_boost_rounds: 43135 Eta:0.05 Max_leaves:6 Feature_fraction:0.5 Lambda_l2:0.6571 Lambda_l1:0.4640 Gamma:0.0468 Bagging_fraction:0.5 Bagging_freq:1

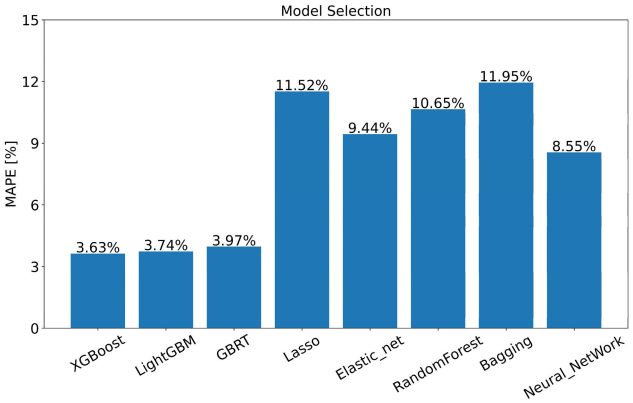

FIGURE 10. Comparison of eight algorithms. Lasso and Elastic net are multi-linear regression algorithms. Bagging is similar to the Random Forest Algorithm.

H. PARAMETERS OF XGBOOST AND LIGHTGBM ALGORITHMS

In this study, we use a small tree depth to avoid overfitting and a large number of base learners to improve the accuracy of prediction. The row sampling and the column sampling strategies are used at the same time to make some disturbance on the input values. Using these sampling strategies to get more different input can effectively prevent over-fitting. If the inputs are always the same values, the model will be easier to be over-fitted. After *early-stopping-rounds* iterations, the algorithm stops iterating when the improvement of accuracy for the test-set is less than the value of the parameter *tol*, which is generally set as 0.0001. The parameters of XGBoost and LightGBM consist of four parts: the complexity of a single base learner, a regularization term, the iteration step size, and the number of iterations. There is no universal way to tune the parameters of machine learning algorithms, most works related to machine learning algorithms are relying on trials and practice. The parameters values selected in this study are shown in Table 3.

TABLE 4. Results of the cross-validation experiment.

Method	Evaluation function	1st Fold	2nd Fold	3rd Fold	4th Fold	5th Fold	Mean
GBRT	MAE	1.0452	1.0434	1.0645	1.03596	1.0308	1.0440
	RMSE	1.5444	1.5421	1.6523	1.5415	1.6303	1.5821
	MAPE	3.76%	3.89%	3.60%	3.73%	4.43%	3.88%
XGBoost	MAE	0.9453	0.9998	1.0205	0.9642	0.9294	0.9718
	RMSE	1.1888	1.1734	1.3846	1.1998	1.2000	1.2293
	MAPE	3.40%	3.74%	3.36%	3.54%	4.19%	3.65%
LightGBM	MAE	1.0053	1.0669	1.0874	1.0393	0.9357	1.0269
	RMSE	1.2659	1.2500	1.5320	1.3106	1.3106	1.3338
	MAPE	3.55%	3.90%	3.58%	3.62%	4.11%	3.75%
Blended Model	MAE	0.503	0.553	0.513	0.563	0.543	**0.535**
	RMSE	0.8797	0.8864	0.8804	0.8790	0.8821	**0.8815**
	MAPE	3.11%	3.21%	3.09%	3.25%	3.29%	3.19%

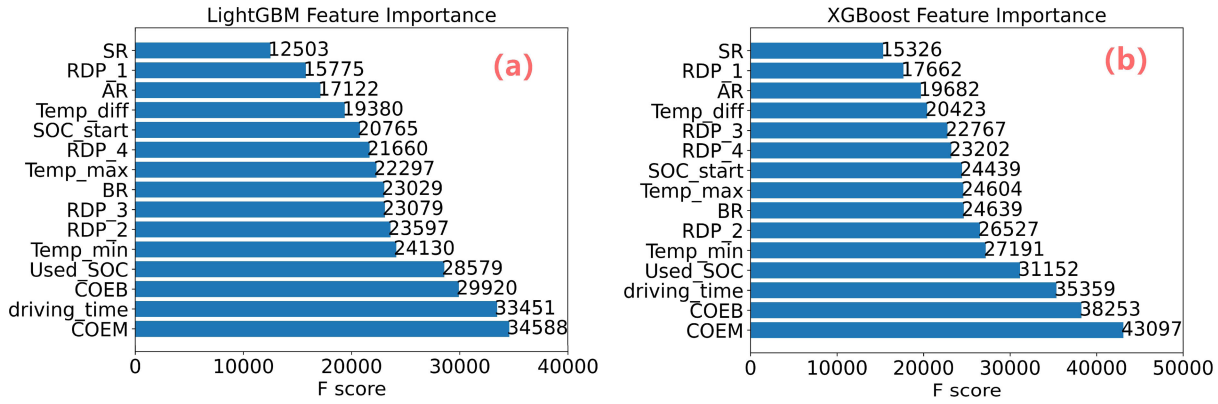

FIGURE 11. Feature importance in the prediction model. Whether in XGBoost or LightGBM, COEM is the most important feature. (a) and (b) illustrate the feature importance rankings for XGBoost and LightGBM algorithms, respectively. F-score denotes the total frequency of using the feature to split data space during the construction of all base trees.

TABLE 5. Results of validation of 120 trips.

MODELS	MAE	RMSE	MAPE	Error range
XGBoost	1.27	1.35	4.12%	[-1.5, 1.5]
LightGBM	1.29	1.32	4.31%	[-1.65,1.65]
Blended Model	0.64	0.94	3.27%	[-0.2,1.20]

IV. EXPERIMENT AND DISCUSSION

Three statistical measures are used to evaluate our method, including the absolute mean error (*MAE*), the root mean squared error (*RMSE*) and mean absolute percentage error (*MAPE*). These values are calculated by formula (15), (16), (17) respectively. *RMSE* and *MAPE* describe how close the prediction value and the true value are. Specifically, *MAPE* describes the relative error, *RMSE* and *MAE* describe the absolute error.

$$MAE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|} \tag{15}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{16}$$

$$MAPE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{y_i}} \tag{17}$$

A. COMPARISON OF MACHINE LEARNING ALGORITHMS

We compared eight different machine learning algorithms including XGBoost, LightGBM, GBRT, Neural network, Random forest, linear regression, etc. for preliminary selection Random forest, linear regression, etc. for preliminary selection.

We randomly split all trips into trainset and test-set with the proportion of 1:1, and then these algorithms were trained and tested on the trainset and the test-set, respectively. The results of comparing these eight algorithms is shown in Fig 10.

All the parameters of these models are tuned to their best states. The parameters for these models are shown in Table 3 and APPENDIX Table A. *Alpha* and *l1-ratio* values for Elastic-net [47] are 1.0 and 0.6 respectively. *Alpha* value for Lasso [48] is 1.0. For Random Forest [49], values of *n_estimators* and *max_depth* are 100 and 6 respectively. The parameters of Neural Network [50] (Multi-Layer Perceptron) are tuned to its best states through the CV-module in scikit-learning (python package).

The comparison results show that XGBoost and LightGBM have lower *MAPE* values than other algorithms in remaining driving range prediction, as shown in Fig. 10. In addition, XGBoost and LightGBM support parallel com-

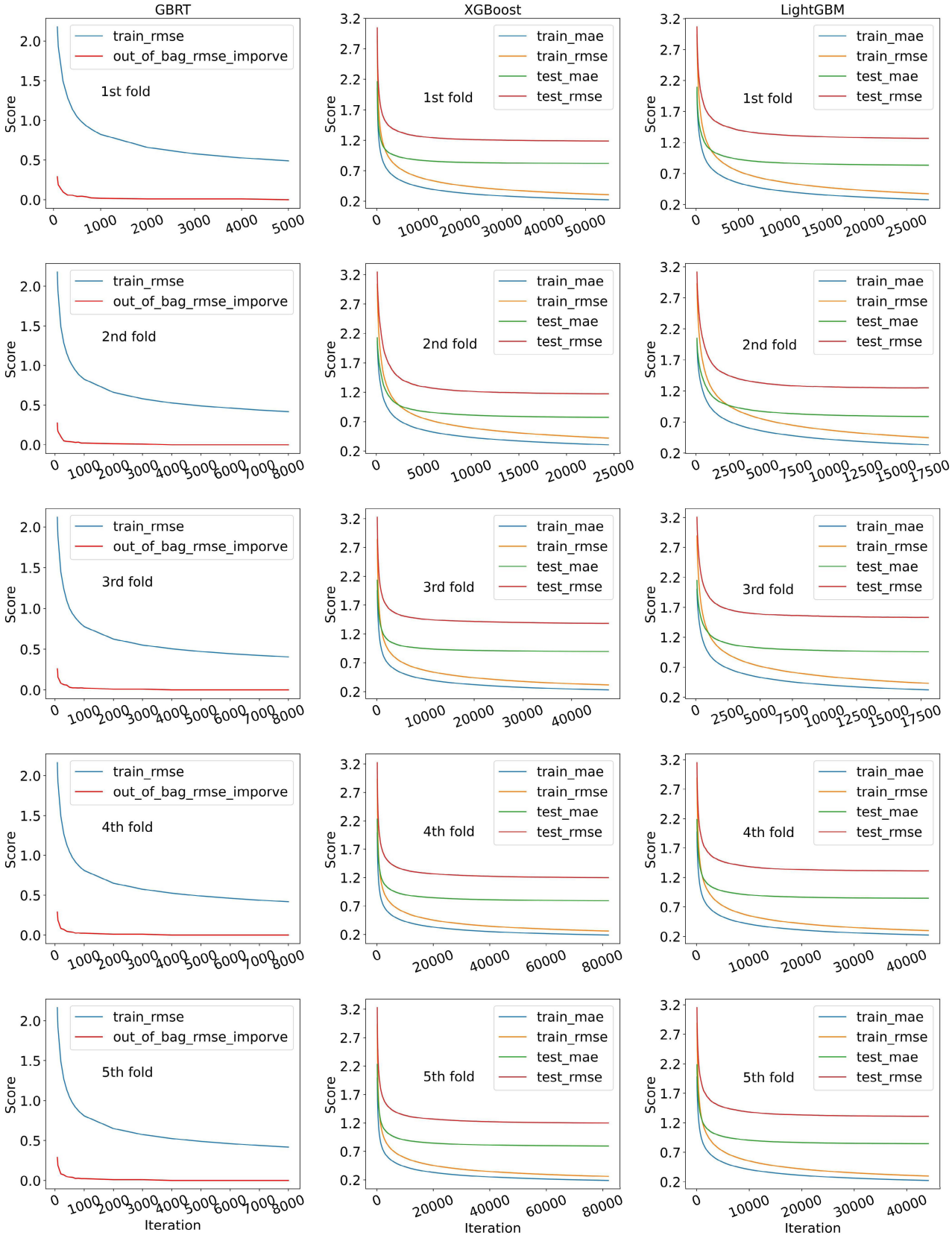

FIGURE 12. RMSE and MAE scores of the training sets and test-sets during iteration. The first 1,000 iterations are not shown.

TABLE 6. Comparison of our results with existing literature works.

METHODS	Model	Features	RMSE	MAE	Error range(km)
Bolovinou [28], 2019[3]	MLR[1]	Speed, EPK (%/km)[4] Road gradient.	---	1.64	---
J, Bi [41], 2019[3]	Nonlinear regression modeling (the quadratic and Weibull Distributions)	Speed, external temperature, DDPSOC[5]	---	---	[-1.4, 1.4]
Shuai Sun [30],2019[3]	GBRT[2]	SOC, MaxT[6], MinT[7], MaxV[8], MinV[9], BV[10], EDT[11],EDV[12],etc	---	0.74	[-1.5, 1.5]
Ours[3] (anchor-free[13])	Blended model	COMB, COME, RDP, BR, SR, AR, etc.	0.94	**0.64**	**[-0.20,1.20]**

[1]Multiple linear regression. [2]Gradient Boost Regression Tree. [3]Collected from one or more vehicles, not the common dataset. [4]Energy consumption rate. [5]Driving distance per SOC. [6]Maximum cell temperature. [7]Minimum cell temperature. [8]Maximum cell voltage. [9]Minimum cell voltage [10]Battery set total voltage [11]Extreme temperature difference [12]Extreme voltage difference. [13]Anchor-free and anchor-based are discussed in Section IV.F

TABLE 7. Results of ablation study of 120 trips.

SOC	COEB	COEM	MAE	MAPE
√			2.351	8.54%
√	√		1.462	4.62%
√	√	√	0.642	3.27%

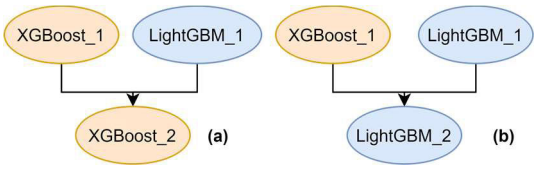

FIGURE 13. Blend strategies.

TABLE 8. Results of comparison of blend strategies.

STRATEGY	MAE	RMSE	MAPE	Error range
(a)	0.64	0.94	3.27%	[-0.2,1.20]
(b)	0.83	1.03	3.52%	[-0.35,1.24]

puting, while GBRT does not support. For further comparing these three algorithms: XGBoost, LightGBM, and GBRT, we conduct a five-fold cross-validation experiment as detailed next.

B. CROSS-VALIDATION EXPERIMENT AND RESULTS

For further comparing GBRT, XGBoost, LightGBM, and our blended model, we conduct a five-fold cross-validation experiment to verify the accuracy and robustness of the machine learning models. The proposed features *COEM* and *COEB* are more important than *Used_SOC,* as demonstrated in Fig 12. Among all the driving features considered, *COEM* is the most important one. The minimum temperature of cells (*Temp_min*) is ranked in fifth, which means that the cells' minimum temperature is a critical factor in range prediction.

The training process is shown in Fig 11. The *RMSE* of the training dataset is always lower than the *RMSE* of the test dataset, but both of them are relatively low.

The results of the cross-validation experiment show that our blended model has the lowest error that *RMES* and *MAE*

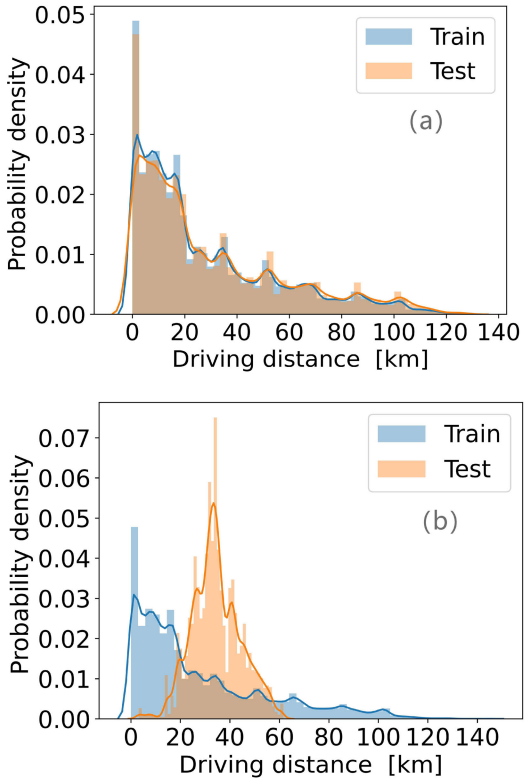

FIGURE 14. Distribution of training and testing data (a) represents that the training data and testing are in the same distribution. (b) represents that the training data and testing data follow the different distributions

are only 0.8815 and 0.535 respectively, as shown in Table 4. The *RMSE* scores of XGBoost and LightGBM are 1.229 and 1.334, respectively, while the *RMSE* score of GBRT is 1.582.

C. VERIFICATION EXPERIMENT FOR THE SECOND STAGE

To verify the second stage of the proposed framework, we performed validation experiments on 120 trips, which are independent of our raw dataset and are collected from the same vehicles. The information about the 120 trips is shown in APPENDIX Fig. A. In the validation experiments, the meaning of T is changed from ''the timestamp when SOC equals zero'' to ''the end timestamp of a trip'' since these

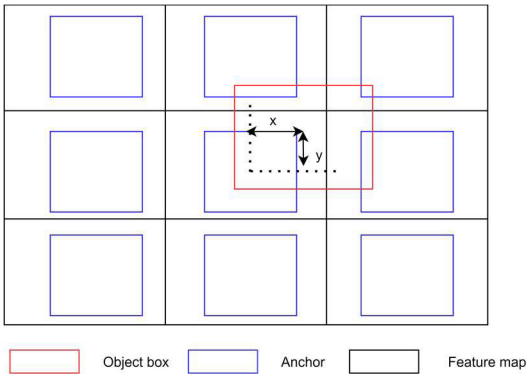

FIGURE 15. Anchor based strategy in 2D object detection

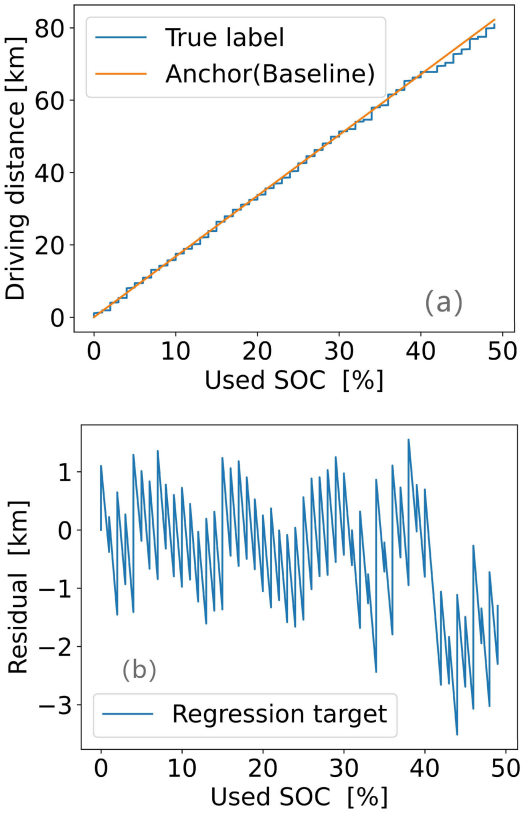

FIGURE 16. Explanation of anchor-based strategy. Subplot (a) represents the anchor (baseline) and the true label. True label is the driving distance calculated through odometer. The anchor (baseline) is the product of DPS and SOC drop. (b) represents the regression targets (denotes as residual), which is the difference between the true label and the anchor.

trips did not exhaust the battery. Therefore, we should predict the remaining distance of trips for comparing with the true values. In real world applications, the meaning of T is "the timestamp when SOC equals zero" for predicting the real remaining range.

The results of the validation experiment on the 120 trips show that the blended model has a small error range of -0.2 to 1.2 km, and it also outperforms XGBoost and LightGBM in MAE and RMSE, as shown in Table 5. Therefore, it is safe to conclude that the blended model has better performance than any other single algorithm. (XGBoost and LightGBM).

TABLE 9. Testing on test-sets that follows different Distributions.

METHODS	DATASET	RMSE	MAE	Error range
Anchor-free	Same[1]	0.94	0.64	[-0.2, 1.2]
	Different[2]	1.33	1.18	[-2.0, 2.2]
Anchor-based	Same[1]	0.73	0.61	[-0.8, 0.8]
	Different[2]	0.75	0.62	[-0.8, 0.8]

1. Training data and testing data follow the same distribution, as shown in Fig 14 (a). 2. Training data and testing data follow the different distributions, as shown in Fig 14 (b).

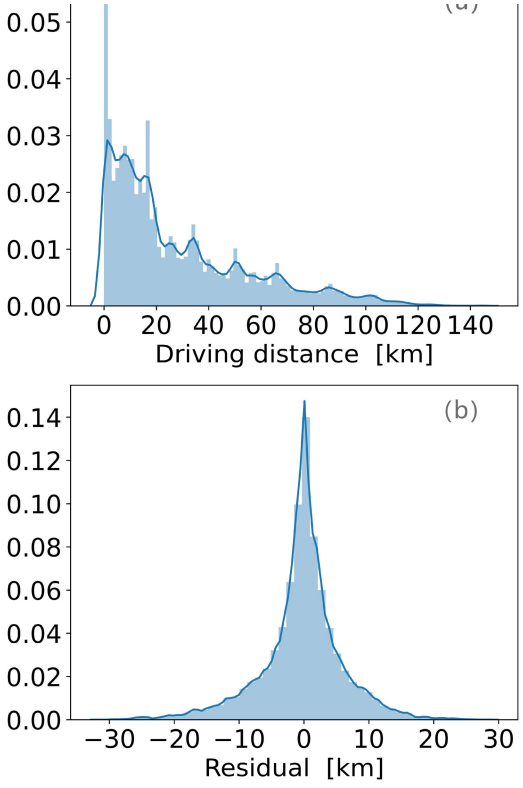

FIGURE 17. The distribution of training label for the anchor-free method (a) and anchor-based method (b). Subplot (a) represents the distribution of driving distance (denoted as true label) calculated through odometer (i.e. odometer(t)- odometer(t_0), t_0 denotes the initial timestamp of a trip and t denotes the current timestamp. (b) is the distribution of anchor-based regression targets, i.e. the residual values of true label and anchor values (baseline). The anchor (baseline) values are the product of *DPS* and SOC drop. SOC drop means $SOC(t_0)$-$SOC(t)$.

Due to the error of the features' prediction model, the results of the validation experiment of 120 trips for the second stage are higher than the results of the cross-validation experiment.

D. COMPARISON WITH EXISTING WORKS

There is no common dataset and benchmark for remaining driving range prediction, which is different from computer vision (e.g. benchmark of object detection—COCO, KITTI, etc.). In previous works [20], [29]–[31], [42], the tests were

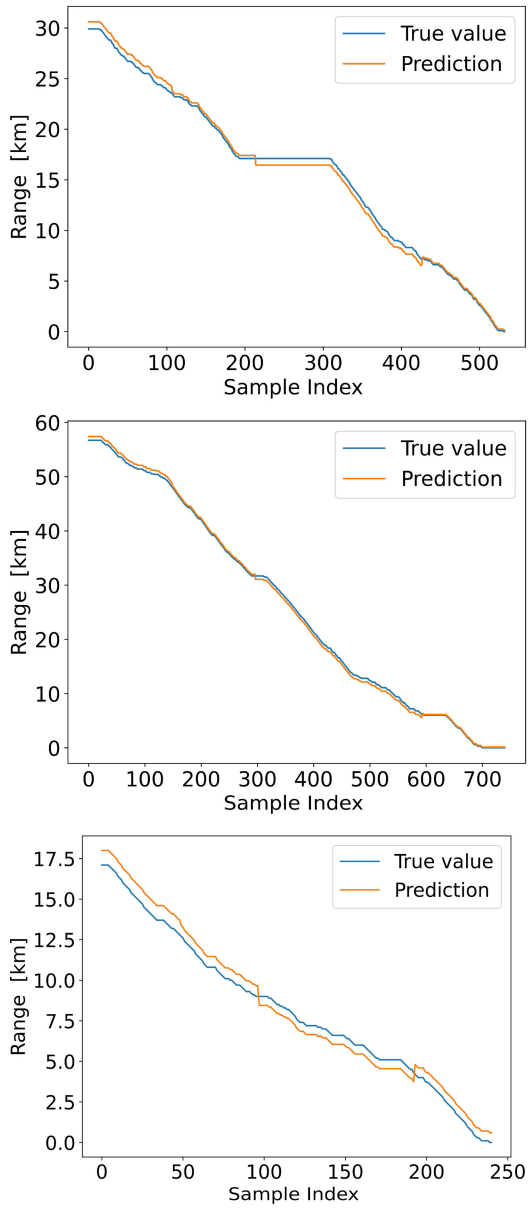

FIGURE 18. **The results of validation experiments on 300 trips. Due to the space limitations, we only represent 3 testing trips selected from 300 testing trips that are independent on the raw dataset and have different distribution from training data. All results of 300 testing trips are shown in Table 9. Timestamp is the sample index. The interval of every sample data is 10 seconds. We did not up-sample the testing data for real simulation.**

done also on their own dataset that are collected from CAN of the vehicle. We, however, note there are some previous studies that used data items [29, 31, 42] similar to ours; so we compared with these methods.

Sun *et al.* [31] and Bi *et al.* [42] only used SOC to represent the remaining energy of battery, while neglecting the degradation of battery. To the contrast, our features prediction model constantly updates the relationship between *COEB* and SOC drop, to overcome the inaccuracy problem of SOC for representing the remaining energy of the battery.

Due to the proposed features (*COEM*, etc.) and the blended model, our method outperforms previous works (i.e. decreasing the MAE by 0.1), as shown in Table 6, in addition to having more robust performances with a smaller error range of [-0.20, 1.20] as compared with previous work.

E. ABLATION STUDIES

To further prove the importance of *COEM* and *COEB*, we conducted an ablation study for the proposed blended model (including the first stage and second stage as described in section III.A and III.B). The blended model was trained on the raw dataset and tested on the 120 trips that are independent of the raw dataset. As shown in Table 7, adding *COEM* and *COEB* to the blended model can decrease the *MAE* and *MAPE* by 1.709 and 5.27 respectively.

We also compared two blend strategies to find the optimal blended model, as shown in Fig 13. The results of the comparison suggest that Strategy (a) has better performance than Strategy (b), as shown in Table 8.

F. ANCHOR-BASED STRATEGY

Motivation: The unbalance of the training data is one of biggest challenges in machine learning studies. Generally speaking, if the training data and testing data follow different distributions, prediction accuracy of machine learning models on testing data will be negatively affected. To explain this phenomenon, a comparison experiment is conducted on two testing data, as shown in Fig 14. In Fig 14 (a) the training data and testing data follow the same distribution, while in Fig 14(b), they follow different distributions. The blended model is trained and tested on the training data and testing data, respectively.

"Anchor" is a professional term in the field of computer vision. Anchor-based strategy is widely used in object detection [51], which uses the anchor boxes (blue) generated in features map to match the ground truth boxes (red) and then calculating the residual values of center coordinates (i.e., x and y, as shown in Fig 15) of the matched anchor boxes and the true boxes as the regression targets for convolution neural networks (CNN).

Inspired by the anchor-based strategy of object detection, we propose an anchor (baseline)-based machine learning method for remaining driving range prediction.

Implementation: specifically, the anchor based strategy converts the true label (driving distance) to the residual values of the true label and anchor as shown in Fig 16.

The anchor (baseline) is the product of the **d**istance **p**er **S**OC (*DPS*, [km/%]) and SOC drop (i.e. SOC(t_0)-SOC(t), t_0 denotes the initial timestamp of a trip and t denotes the current timestamp.). *DPS* is calculated through more than 2000 trips, which can be regarded as a statistics value. In our dataset, *DPS* equals 1.6775. The distribution of anchor-based training label (regression targets) and anchor-free training label are presented in Fig 17 (a) and (b).

Comparative studies are conducted to prove the effectiveness of the anchor-based strategy. Specifically, two sets of

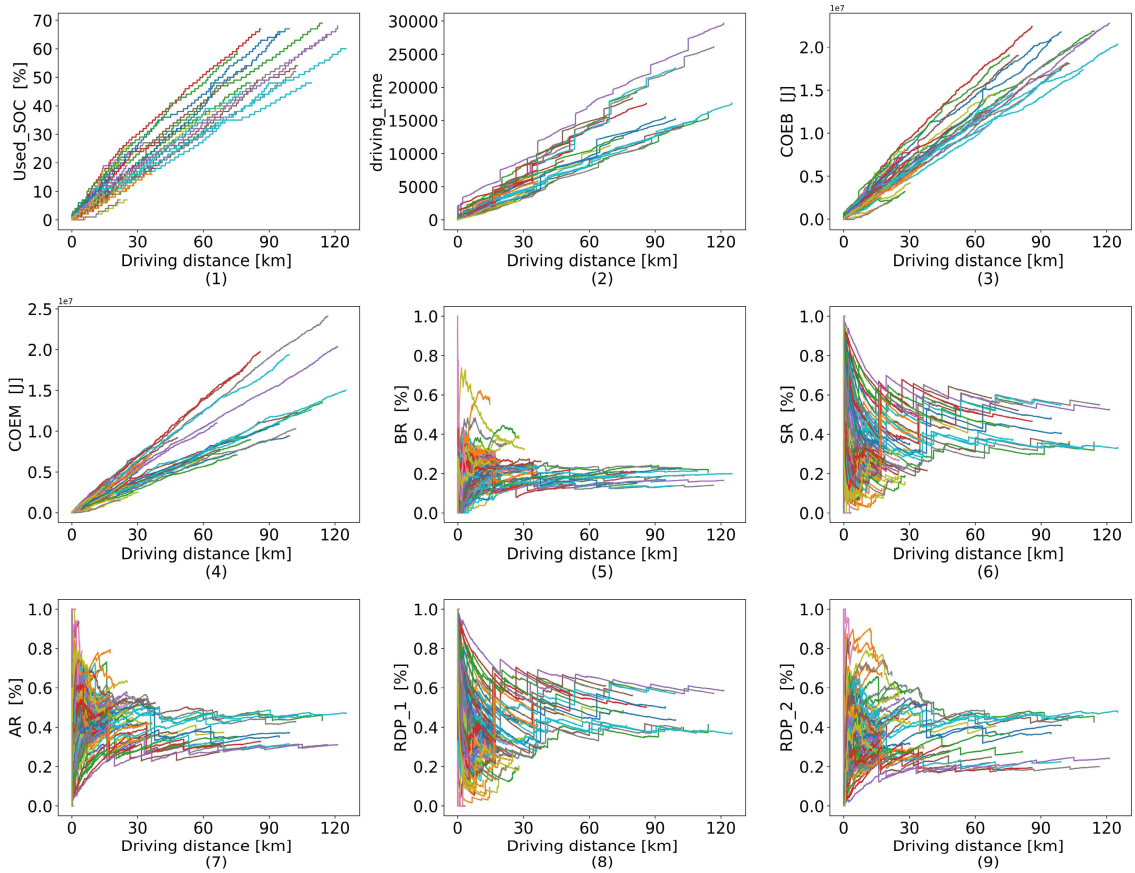

FIGURE 19. Validation data collect from the same vehicle. The validation data consists of 120 trips that are independent of the raw dataset. Every subplot contains 120 trips, and the different color represent different trips. Due to the space limitation, some features are not shown in this figure.

data are used to test the anchor-based and the anchor-free methods: 1) the training data and testing data that follow the same distribution are denoted as Set_1 (see Fig 14 (a)); and 2) the training data and testing data which follow different distributions are denoted as Set_2(see Fig 14 (b)). The results of the comparison, as listed in Table 9, suggest that the performance of the anchor-free model deteriorates on testing data that follows different distribution from the training data; while the anchor-based model has more robust performances (i.e. error range of [-0.8, 0.8] km) on both testing data (Fig 14 (a) and Fig 14 (b)), proving the effectiveness of anchor-based strategy. The results of validation experiment for anchor-based blended model is presented in Fig 18.

V. CONCLUSION

Accurate prediction of the remaining driving range is important to EV drivers. In this paper, a two-stage framework for the remaining driving range prediction of EVs has been proposed by merging two advanced machine learning algorithms of XGBoost and LightGBM. The results of experiments show that the blended model has a smaller error range of $[-0.8, 0.8]$ and a lower RMSE of 0.75 as compared to the error range of [1.4, 1.4] in previous works. Besides, we use the cumulative output energy of the motor and the batteries to reflect the battery degradation, and use driving patterns related features to represent the energy consumption level. These features, which are proposed for the first time in our method, can effectively improve the accuracy of remaining driving range prediction. Adding the cumulative output energy of the motor and the batteries to the blended model can decrease the *MAPE* by 5.27%. Anchor-based strategy was proposed for the first time in our method, which converts the true label (driving distance) to the residual values between the baseline and the true label for training. Anchor-based strategy adopted in remaining driving range prediction can solve the unbalance distribution of training data and achieve high performance (i.e. low error range of [-0.8, 0.8]) on the testing data that follows different distribution from training data.

The results of the comparison between the anchor-based and anchor-free strategy suggest that the former always has a stable error range of [-0.8, 0.8] for testing data that either follows the same or different distribution from the training data; on the other hand, the performance of the anchor-free blended model deteriorate (i.e. an error range of [-2.0, 2.2]) on testing data that follows the different distribution from the training data.

TABLE 10. Parameters values used in the machine learning algorithms. Default values are not listed.

Neural network Parameters	**Bagging** Parameters	**Lasso** Parameters	**Elastic-net** Parameters	**Random forest** Parameters
hidden_layer_sizes:1500 activation:relu solver:adam alpha:0.0001	base_estimator:default n_estimators:2200 max_samples:0.5 max_features:0.45	alpha:1.0 fit_intercept:True Normalize:true max_iter:3500 tol:3e-4	alpha:0.85 L1 ratio:0.8 Fitintercept:true Max-iter:3500 tol: 3e-4	n_estimators:5500 criterion:mse max_depth:3 min_samples_split:2 max_features:auto min_samples_leaf:1

APPENDIX

The information on testing data.

Computation Platform Details: The computing platform is *Ubuntu 18.04*, the programming language is *python*. The CPU of our platform is i9-9900KF@ 3.60GHz∗16, the RAM is 32GB. The prediction process in our platform only needs 32MB RAM and 0.05 second.

Tuning of Parameters: There is a CV-module in machine learning package (scikit-learn; website is https://scikit-learn.org/stable/), which can be used to search the best combination of parameters in the parameter space. Note the parameter space is manually defined, which can be understood as lists of values for parameters. Specifically, the CV-module will try every possible combination of the parameters and compare all results to find the optimal combination of parameters.

ACKNOWLEDGMENT

The authors would like to acknowledge the National Big Data Alliance of New Energy Vehicles for their contribution in providing data; the 111 Project (B17034) for their contribution in providing computing resources; and the Innovative Research Team Development Program of the Ministry of Education of China (IRT_17R83) for its support to their team.

REFERENCES

[1] Y. Tao, M. Huang, Y. Chen, and L. Yang, "Orderly charging strategy of battery electric vehicle driven by real-world driving data," *Energy*, vol. 193, Feb. 2020, Art. no. 116806, doi: 10.1016/j.energy.2019.116806.

[2] Y. Luo, G. Feng, S. Wan, S. Zhang, V. Li, and W. Kong, "Charging scheduling strategy for different electric vehicles with optimization for convenience of drivers, performance of transport system and distribution network," *Energy*, vol. 194, Mar. 2020, Art. no. 116807, doi: 10.1016/j.energy.2019.116807.

[3] M. Knez, G. K. Zevnik, and M. Obrecht, "A review of available chargers for electric vehicles: United states of America, European Union, and Asia," *Renew. Sustain. Energy Rev.*, vol. 109, pp. 284–293, Jul. 2019, doi: 10.1016/j.rser.2019.04.013.

[4] D. D. Tran, M. Vafaeipour, M. El Baghdadi, R. Barrero, and J. Van Mierlo, "Thorough state-of-the-art analysis of electric and hybrid vehicle powertrains: Topologies and integrated energy management strategies," *Renew. Sust. Energ. Rev.*, vol. 119, Mar. 2019, Art. no. 109596, doi: 10.1016/j.rser.2019.109596.

[5] C. She, Z. Wang, F. Sun, P. Liu, and L. Zhang, "Battery aging assessment for real-world electric buses based on incremental capacity analysis and radial basis function neural network," *IEEE Trans. Transport. Electrific.*, vol. 16, no. 5, pp. 3345–3354, May 2020, doi: 10.1109/TII.2019.2951843.

[6] S. Paul, C. Diegelmann, H. Kabza, and W. Tillmetz, "Analysis of ageing inhomogeneities in lithium-ion battery systems," *J. Power Sources*, vol. 239, pp. 642–650, Oct. 2013, doi: 10.1016/j.jpowsour.2013.01.068.

[7] Q. Wang, Z. Wang, L. Zhang, P. Liu, and Z. Zhang, "A novel consistency evaluation method for series-connected battery systems based on real-world operation data," *IEEE Trans. Transport. Electrific.*, early access, Aug. 20, 2020, doi: 10.1109/TTE.2020.3018143.

[8] S. F. Schuster, M. J. Brand, P. Berg, M. Gleissenberger, and A. Jossen, "Lithium-ion cell-to-cell variation during battery electric vehicle operation," *J. Power Sources*, vol. 297, pp. 242–251, Nov. 2015, doi: 10.1016/j.jpowsour.2015.08.001.

[9] L. Zhang, W. Fan, Z. Wang, W. Li, and D. U. Sauer, "Battery heating for lithium-ion batteries based on multi-stage alternative currents," *J. Energy Storage*, vol. 32, Dec. 2020, Art. no. 101885, doi: 10.1016/j.est.2020.101885.

[10] L. Huang, Z. Zhang, Z. Wang, L. Zhang, X. Zhu, and D. D. Dorrell, "Thermal runaway behavior during overcharge for large-format lithium-ion batteries with different packaging patterns," *J. Energy Storage*, vol. 25, Oct. 2019, Art. no. 100811, doi: 10.1016/j.est.2019.100811.

[11] X. Feng, J. Sun, M. Ouyang, F. Wang, X. He, L. Lu, and H. Peng, "Characterization of penetration induced thermal runaway propagation process within a large format lithium ion battery module," *J. Power Sources*, vol. 275, pp. 261–273, Feb. 2015, doi: 10.1016/j.jpowsour.2014.11.017.

[12] J. Ye, H. Chen, Q. Wang, P. Huang, J. Sun, and S. Lo, "Thermal behavior and failure mechanism of lithium ion cells during overcharge under adiabatic conditions," *Appl. Energy*, vol. 182, pp. 464–474, Nov. 2016, doi: 10.1016/j.apenergy.2016.08.124.

[13] Q. Wang, P. Ping, X. Zhao, G. Chu, J. Sun, and C. Chen, "Thermal runaway caused fire and explosion of lithium ion battery," *J. Power Sources*, vol. 208, pp. 210–224, Jun. 2012, doi: 10.1016/j.jpowsour.2012.02.038.

[14] H. Zhang, X. Song, T. Xia, M. Yuan, Z. Fan, R. Shibasaki, and Y. Liang, "Battery electric vehicles in Japan: Human mobile behavior based adoption potential analysis and policy target response," *Appl. Energy*, vol. 220, pp. 527–535, Jun. 2018, doi: 10.1016/j.apenergy.2018.03.105.

[15] L. Noel, G. Zarazua de Rubens, B. K. Sovacool, and J. Kester, "Fear and loathing of electric vehicles: The reactionary rhetoric of range anxiety," *Energy Res. Social Sci.*, vol. 48, pp. 96–107, Feb. 2019, doi: 10.1016/j.erss.2018.10.001.

[16] *National Big Data Alliance of New Energy Vehicles*. Website. [Online]. Available: http://www.ndanev.com/

[17] K.-H. Shin, C.-S. Jin, J.-Y. So, S.-K. Park, D.-H. Kim, and S.-H. Yeon, "Real-time monitoring of the state of charge (SOC) in vanadium redox-flow batteries using UV–Vis spectroscopy in operando mode," *J. Energy Storage*, vol. 27, Feb. 2020, Art. no. 101066, doi: 10.1016/j.est.2019.101066.

[18] X. Tang, Y. Wang, K. Yao, Z. He, and F. Gao, "Model migration based battery power capability evaluation considering uncertainties of temperature and aging," *J. Power Sources*, vol. 440, Nov. 2019, Art. no. 227141, doi: 10.1016/j.jpowsour.2019.227141.

[19] X. Tang, K. Liu, X. Wang, F. Gao, J. Macro, and W. D. Widanage, "Model migration neural network for predicting battery aging trajectories," *IEEE Trans. Transport. Electrific.*, vol. 6, no. 2, pp. 363–374, Jun. 2020, doi: 10.1109/TTE.2020.2979547.

[20] H. A. Yavasoglu, Y. E. Tetik, and K. Gokce, "Implementation of machine learning based real time range estimation method without destination knowledge for BEVs," *Energy*, vol. 172, pp. 1179–1186, Apr. 2019, doi: 10.1016/j.energy.2019.02.032.

[21] G. M. Fetene, S. Kaplan, S. L. Mabit, A. F. Jensen, and C. G. Prato, "Harnessing big data for estimating the energy consumption and driving range of electric vehicles," *Transp. Res. D, Transp. Environ.*, vol. 54, pp. 1–11, Jul. 2017, doi: 10.1016/j.trd.2017.04.013.

[22] H. Guo, S. Lu, H. Hui, C. Bao, and J. Shangguan, "Receding horizon control-based energy management for plug-in hybrid electric buses using a predictive model of terminal SOC constraint in consideration of stochastic vehicle mass," *Energy*, vol. 176, pp. 292–308, Jun. 2019, doi: 10.1016/j.energy.2019.03.192.

[23] X. Tang, F. Gao, C. Zou, K. Yao, W. Hu, and T. Wik, "Load-responsive model switching estimation for state of charge of lithium-ion batteries," *Appl. Energy*, vol. 238, pp. 423–434, Mar. 2019, doi: 10.1016/j.apenergy.2019.01.057.

[24] K. Liu, J. Wang, T. Yamamoto, and T. Morikawa, "Exploring the interactive effects of ambient temperature and vehicle auxiliary loads on electric vehicle energy consumption," *Appl. Energy*, vol. 227, pp. 324–331, Oct. 2018, doi: 10.1016/j.apenergy.2017.08.074.

[25] K. Liu, T. Yamamoto, and T. Morikawa, "Impact of road gradient on energy consumption of electric vehicles," *Transp. Res. D, Transp. Environ.*, vol. 54, pp. 74–81, Jul. 2017, doi: 10.1016/j.trd.2017.05.005.

[26] R. Galvin, "Energy consumption effects of speed and acceleration in electric vehicles: Laboratory case studies and implications for drivers and policymakers," *Transp. Res. D, Transp. Environ.*, vol. 53, pp. 234–248, Jun. 2017, doi: 10.1016/j.trd.2017.04.020.

[27] 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, May 2017, doi: 10.3390/en10050608.

[28] C.-H. Lee and C.-H. Wu, "A novel big data modeling method for improving driving range estimation of EVs," *IEEE Access*, vol. 3, pp. 1980–1993, 2015, doi: 10.1109/access.2015.2492923.

[29] A. Bolovinou, I. Bakas, A. Amditis, F. Mastrandrea, and W. Vinciotti, "Online prediction of an electric vehicle remaining range based on regression analysis," in *Proc. IEEE Int. Electr. Vehicle Conf. (IEVC)*, Dec. 2014, pp. 1–8.

[30] J. Bi, Y. Wang, S. Shao, and Y. Cheng, "Residual range estimation for battery electric vehicle based on radial basis function neural network," *Measurement*, vol. 128, pp. 197–203, Nov. 2018, doi: 10.1016/j.measurement.2018.06.054.

[31] S. Sun, J. Zhang, J. Bi, and Y. Wang, "A machine learning method for predicting driving range of battery electric vehicles," *J. Adv. Transp.*, vol. 2019, pp. 1–14, Jan. 2019, doi: 10.1155/2019/4109148.

[32] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[33] G. L. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Y. Liu, "LightGBM: A highly efficient gradient boosting decision tree," in *Proc. 31st Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA, Dec. 2017, pp. 1–9.

[34] X. Gu, Y. Han, and J. Yu, "A novel lane-changing decision model for autonomous vehicles based on deep autoencoder network and XGBoost," *IEEE Access*, vol. 8, pp. 9846–9863, 2020, doi: 10.1109/ACCESS.2020.2964294.

[35] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and XGboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018, doi: 10.1109/ACCESS.2018.2818678.

[36] M. Al-Rakhami, A. Gumaei, A. Alsanad, A. Alamri, and M. M. Hassan, "An ensemble learning approach for accurate energy load prediction in residential buildings," *IEEE Access*, vol. 7, pp. 48328–48338, 2019, doi: 10.1109/ACCESS.2019.2909470.

[37] Y. Qu, Z. Lin, H. Li, and X. Zhang, "Feature recognition of urban road traffic accidents based on GA-XGBoost in the context of big data," *IEEE Access*, vol. 7, pp. 170106–170115, 2019, doi: 10.1109/ACCESS.2019.2952655.

[38] T. Halmeaho, P. Rahkola, K. Tammi, J. Pippuri, A.-P. Pellikka, A. Manninen, and S. Ruotsalainen, "Experimental validation of electric bus powertrain model under city driving cycles," *IET Electr. Syst. Transp.*, vol. 7, no. 1, pp. 74–83, Mar. 2017, doi: 10.1049/iet-est.2016.0028.

[39] 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, Apr. 2016, doi: 10.1016/j.apenergy.2016.01.097.

[40] K. N. Genikomsakis and G. Mitrentsis, "A computationally efficient simulation model for estimating energy consumption of electric vehicles in the context of route planning applications," *Transp. Res. D, Transp. Environ.*, vol. 50, pp. 98–118, Jan. 2017, doi: 10.1016/j.trd.2016.10.014.

[41] J. G. Hayes, R. P. R. de Oliveira, S. Vaughan, and M. G. Egan, "Simplified electric vehicle power train models and range estimation," in *Proc. IEEE Vehicle Power Propuls. Conf.*, Sep. 2011, pp. 1–5, doi: 10.1109/VPPC.2011.6043163.

[42] J. Bi, Y. Wang, Q. Sai, and C. Ding, "Estimating remaining driving range of battery electric vehicles based on real-world data: A case study of beijing, China," *Energy*, vol. 169, pp. 833–843, Feb. 2019, doi: 10.1016/j.energy.2018.12.061.

[43] S. Ghemawat, H. Gobioff, and S.-T. Leung, "The Google file system," *ACM SIGOPS Operating Syst. Rev.*, vol. 37, no. 5, pp. 29–43, Dec. 2003, doi: 10.1145/1165389.945450.

[44] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," *Commun. ACM*, vol. 51, no. 1, pp. 107–113, Jan. 2008, doi: 10.1145/1327452.1327492.

[45] F. Chang, J. Dean, S. Ghemawat, W. C. Hsieh, D. A. Wallach, M. Burrows, T. Chandra, A. Fikes, and R. E. Gruber, "Bigtable: A distributed storage system for structured data," *ACM Trans. Comput. Syst.*, vol. 26, no. 2, pp. 1–26, Jun. 2008, doi: 10.1145/1365815.1365816.

[46] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, 2002, doi: 10.1016/S0167-9473(01)00065-2.

[47] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Statist. Softw.*, vol. 33, no. 1, pp. 1–22, 2010.

[48] S. J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky, "An interior-point method for large-scale L1-regularized least squares," *IEEE J. Sel. Topics Signal Process.*, vol. 1, no. 4, pp. 606–617, Dec. 2008, doi: 10.1109/JSTSP.2007.910971.

[49] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001, doi: 10.1023/A:1010933404324.

[50] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[51] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multiBox detector," presented at the 14th Eur. Conf. Comput. Vis. (ECCV), Amsterdam, The Netherlands, 2016, doi: 10.1007/978-3-319-46448-0_2.

LIANG ZHAO was born in Yulin, Shaanxi, China, in 1996. He received the B.S. degree in vehicle engineering from Chang'an University, Xi'an, China, in 2018.

Since 2018, he was a Research Assistant with the Hubei Research Center for New Energy and Intelligent Connected Vehicles.

He is currently a Data Scientist with the Wuhan University of Technology, where he is involved in the application of vehicle driving data. His current research interests include data mining, 3-D object detection in self-driving cars, development of machine learning algorithms, and application of deep learning in the self-driving cars.

WEI YAO was born in Linchuan, Jiangxi, China, in 1980. She received the B.S. and M.S. degrees from the Huazhong University of Science and Technology in 2002 and 2005, respectively, and the Ph.D. degree from Wuhan University in 2017. She has been an Assistant Professor with the Wuhan University of Technology. Her research interests include machine learning and data analysis.

YU WANG received the B.S. degree in energy and power engineering from the Wuhan University of Technology in 2006, the M.S. degree in energy and power engineering from Tsinghua University in 2009, and the Ph.D. degree in energy and power engineering from the King Abdullah University of Science and Technology (KAUST) in 2013. From 2013 to 2015, he was a Research Specialist with the Department of Mechanical Engineering, KAUST. Since 2015, he has been a Professor with the Department of Energy and Power Engineering, Wuhan University of Technology. He has authored more than 30 articles. His research interests include renewable and sustainable energy, electric vehicle, and combustion science.

JIE HU received the B.S., M.S., and Ph.D. degrees in energy and power engineering from the Wuhan University of Technology, in 2006, 2009, and 2011, respectively. From 2011 to 2013, he was a Research Specialist with SGMW Company. Since 2013, he has been an Associate Professor with the Department of Automobile Engineering, Wuhan University of Technology.