A Prediction Method of Driving Range Based on LSTM Combined with Causal Convolution

Zhe Zuo *, Ning Xu, ZhenYu Zhang, Yuheng Yan, Weilong Lv, and Ziyang Xu

School of Mechanical Engineering, Beijing Institute of Technology,
No. 5 Zhongguancun South Street, Haidian District, Beijing 100081, China
E-mail: zuzeus@bit.edu.cn

Abstract. Accurate prediction of driving range of electric vehicles (EVs) is critically important to help optimize energy management. In this paper, a novel method to perform accurate driving range prediction is investigated using long short-term memory(LSTM) recurrent neural network combined with convolution layers. The data used in this paper are obtained from the National Monitoring and Management Platform for New energy vehicles. First, the data cleaning is applied to deal with abnormal values and missing values. Then, the datasets are constructed by the sliding window method. Finally, the LSTM model is built and the convolution layers are connected to the model. Dropout method is also introduced to prevent the model from overfitting. The results show that the proposed method can accurately predict the driving range, and the average RMSPE 0.099.

Keywords: Driving range prediction, LSTM, Deep Learning

1. INTRODUCTION

In recent years, electric vehicles have attracted widespread attention due to “zero pollution”. However, due to the short driving range of electric vehicles, it is easy for drivers to have range anxiety during the driving, which in some way restricts the popularization of electric vehicles. The changes in the traffic environment and driving conditions will result in a large gap between the ideal estimated driving range and the actual driving range of electric vehicles. Therefore, using a data-driven method based on massive actual driving data to provide accurate prediction of driving range timely is important to the further development of electric vehicles. And the model built in this paper is an end-to-end model, which can help reduce the complexity of building models and reduce the impact of insufficient understanding of the theoretical system of the batteries. The accuracy of driving range prediction is the key to mitigating range anxiety, improving energy utilization, and optimizing energy management of electric vehicles. In recent years, many researchers have proposed a variety of methods to predict the driving range.

Cunningham I et al. [1] used the "fuzzy transformation", which is a relatively new development of the model-free method to predict the remaining mileage of electric vehicles online. Liu G et al.[2] presented a battery energy prediction (EP) method based on the predictive control theory, in which a coupled prediction of future battery state variation, battery model parameter change, and voltage response, is implemented on the ERDE prediction horizon, and the ERDE is subsequently accumulated and real-timely optimized. The results show that the EP method provides better accuracy than the traditional DC method. PAEP can reduce the E-RDE error by more than 90% and ensure that the relative energy prediction error is below 2%.

With the increasing number of electric vehicles, there are more and more data characterizing actual driving conditions of electric vehicles. Massive data contain rich value. Using methods of big data analysis and machine learning to predict the driving range can make up for the shortcomings of traditional modeling methods. Factors that are difficult to be considered in the traditional ways are reflected in the data. Support vector machine (SVM) is one of frequently used methods. SVM is a machine learning method based on statistical learning theory that can satisfy the Vapnik-Chervonenkis Dimension theory and structural risk minimum principle under finite sample conditions, which has strong generalization ability, global optimization and fast calculation speed [3]. Based on machine learning, many researchers have conducted related research. Wang Z et al [4] presented a novel prediction method based on a least squares support vector machine (LSSVM) model with parameters γ and σ2 optimized by particle swarm optimization (PSO). Days, temperature, depth of discharge (DOD) of battery pack are used for training model and AARE of testing data is 5.99%.

Recently, deep learning has attracted a lot of attention, and researchers gradually apply recurrent neural networks (RNN) to the related research fields in electric vehicles. Compared with SVM, RNN can effectively learn long term and short term dependence information in time series, which is more advantageous in dealing with time series data [5]. On the basis of RNN, LSTM adds the process of filtering historical data, which effectively solves the problem of gradient disappearance and gradient explosion [6]. Hong J et al. [7] investigated a new deep-learning-enabled method to perform accurate synchronous multi-parameter prediction for battery systems using a long short-term memory (LSTM) recurrent neural network. Hong J et al. [8] applied LSTM to voltage prediction and fault prognosis of the battery system. Li X et al. [9] presented a novel hybrid Elman-LSTM method for battery remaining useful life prediction by combining the empirical model decomposition algorithm and long short-term memory and Elman neural networks.

Inspired by the above-mentioned research, considering that the data used in driving range prediction is high-dimensional, large-scale, and time-related, this paper chooses to use long and short-term memory neural networks for modeling. The data used are divided into a training set and a test set. After cleaning the data, the sliding window method is used to construct the data set. We use the training set to train the LSTM model, and use the test set to evaluate the prediction accuracy of the model.

2. BACKGROUND

2.1. Causal Convolution

The causal convolution is a kind of one-dimensional convolution designed for time-series problem. The basic principle of causal convolution is shown in (1).

$$\boldsymbol{p}(\boldsymbol{x}) = \prod_{t=1}^{T} \boldsymbol{p}(\boldsymbol{x_t}|\boldsymbol{x_{t-n}}, \dots\dots, \boldsymbol{x_{t-1}}) \qquad (1)$$

The above formula represents the probability $\mathbf{p}(\mathbf{x})$ of obtaining the information of time step t when the sequence information $(\boldsymbol{x_{t-n}}, \dots\dots, \boldsymbol{x_{t-1}})$ is input. The purpose if model training is to maximize the probability of the real value $\boldsymbol{x_t}$ of the output. The basic principle of causal revolution is shown in **Fig. 1**.

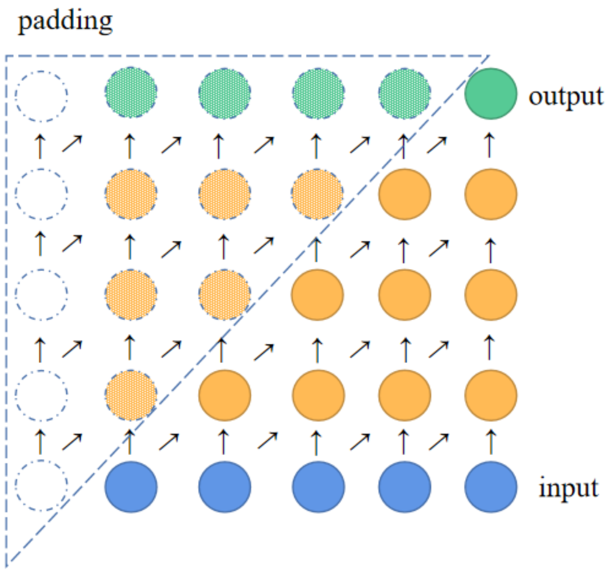


Fig. 1 A stack of causal convolutional layers

In the above figure, blue dots represent the input variables, yellow dots represent the intermediate variables, and green dots represent the output results. It can be seen that the output of each causal convolutional layer is determined by the pervious time's layer. The area surrounded by the dashed triangles shows the padding method, which can make the output has the same dimension with the input.

2.2. RNN

RNN is a type of neural network dedicated to processing sequence data. In ANN, the current output of the system is only related to the current input, while in the RNN, the current output of the system is not only related to the current input, but also related to the previous output. The specific theory is that the network memorizes the previous information and applies it to the calculation of the current output, which means that the nodes between the hidden layers are no longer disconnected, and the input of the hidden layer includes not only the input layer but also the output of the hidden layer at previous moments.

The structure of the RNN is shown in **Fig. 2**, x is a sequence of length T, where x is the input layer, $x^{(t)}$ is the input of x at time t, and h is the hidden layer, $h^{(t)}$ is the state of the hidden layer at time t, o is the output layer, $\mathrm{o}^{(t)}$ is the output of o at time t, $L^{(t)}$ represents the loss function at time t, $\mathrm{y}^{(t)}$ is the true output at time t. W、U and V are the connection weights of the input layer, the hidden layer, and the output layer respectively, which are shared by the entire RNN. Meanwhile, as can be seen from **Fig. 2**, the state of the hidden layer $h^{(t)}$ is related not only to the current time's input $x^{(t)}$ but also to the state $h^{(t-1)}$ of the hidden layer at the previous time.

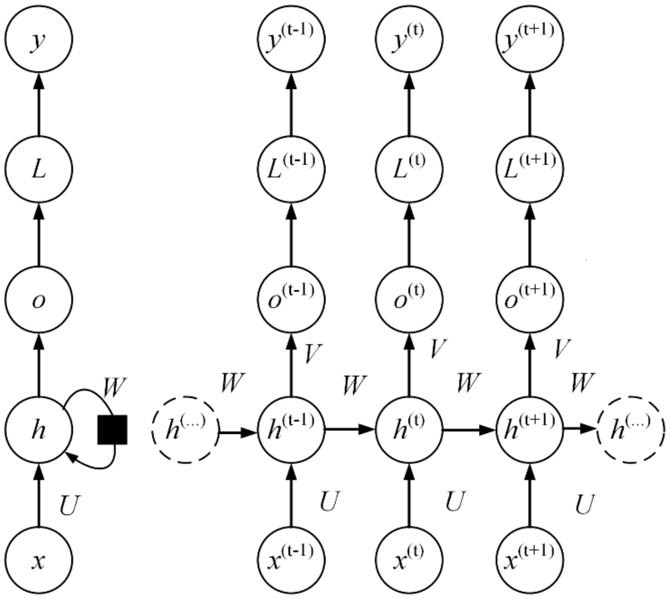


Fig. 2 The structure of RNN

The propagation process is shown in (2), (3), (4) and (5).

$$h^{(t)} = \alpha(b + Uh^{(t-1)} + Wx^{(t)}) \qquad (2)$$

$$o^{(t)} = c + Vh^{(t)} \qquad (3)$$

$$\hat{y}^{(t)} = \delta(o^{(t)}) \tag{4}$$

$$L^{(t)} = \hat{y}^{(t)} - y^{(t)} \tag{5}$$

Where α and δ are activation functions, $\hat{y}^{(t)}$ is the predicted output at time t, and b and c are the intercepts of the linear relationship respectively.

2.3. LSTM

LSTM is a variant of RNN that effectively learns long-term dependencies. All RNN share a chained form of a repetitive neural network module. In a standard RNN, this duplicate module has only a very simple structure, such as a single tanh layer. However , in the LSTM, there are four layers that interact in a special way.

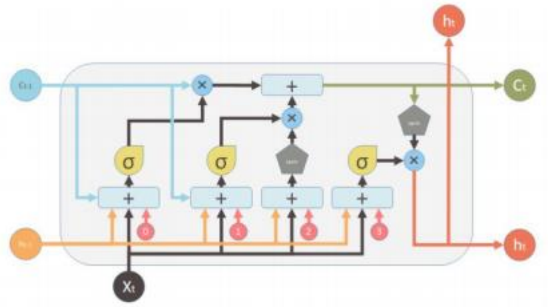


Fig. 3 The structure of LSTM

Fig. 3 shows the structure of the LSTM. The working process is shown in (6), (7) and (8).

$$i = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{6}$$

$$f = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{7}$$

$$o = \sigma(W_o h_{t-1} + U_0 x_t + b_0) \tag{8}$$

Where i , f , o are the input gate, the forget gate and the output gate respectively. The input gate determines the current input information and defines how much the current input information can pass. The forget gate determines the historical information and defines how much the last state output information can pass. The output gate determines the output information and defines how much of the current output information can be received by the next state. The activation function δ make the output value of each gate to be between 0 and 1. For the forget gate, when f is 0, it means to completely discard the information of the previous state; when f is 1, it means to completely retain the previous state information.

3. METHOD

Data preprocessing is very important in the process of building the model, for the high-quality data are the basis of the model's accuracy. It deals with the problem that data may be missing or there are some noise in the data when being collected. And there are also some data that are not necessary to be used when building the model ,which means they need to be removed.In this paper,data cleaning ,data transformation and data reduction are used to preprocess the data.

3.1. Data Cleaning

The data in this paper are obtained from the actual operating data of a certain electric vehicle in Beijing released by the National Monitoring and Management Platform for New energy vehicles. The data collected include the followings: collected time of each piece of data, total voltage of the batteries, total current of the batteries, state of charge, maximum temperature of the cell, minimum temperature of the cell, input voltage of motor controller, DC bus current of motor controller and accumulated mileage. Attributes of the original data are shown in Table 1.

Table. 1 Attributes of the original data

name	symbol	units	meaning
time	T	s	collected time
total_voltage	V_t	V	total voltage of the battery
total_current	I_t	A	total current of the battery
soc	S		state of charge
temp_max	T_b	℃	maximum temperature of the cell
temp_min	Ts	℃	minimum temperature of the cell
motor_voltage	V_m	V	input voltage of motor controller
motor_current	I_m	A	DC bus current of motor controller
mileage	M	km	accumulated mileage

Box plot is a kind of statistical chart to show the distribution of data. As shown in **Fig. 4,** there are some abnormal values. The abnormal data mainly include missing values and outliers. It should be noted that the data in the Figure. 4 has been standardized before plotted, and the standardization method is described in section 3.3. The abnormal data will directly influence the accuracy of the neural network.

For missing values, linear interpolation is used to fill in. The formula is show in (9).

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0) \tag{9}$$

Where (x_0, y_0) and (x_1, y_1) are the known points without missing values, and (x, y) is the missing point.

Outliers are processed by either using mean value to replace or simply being eliminated.

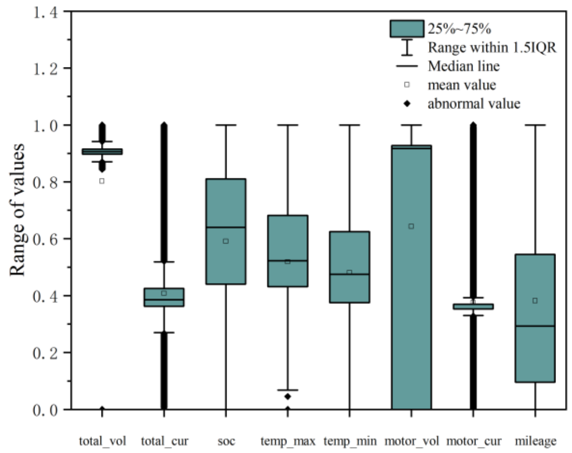


Fig. 4 Box plot

3.2. Feature Engineering and Build the Dataset

In this paper, feature engineering is applied to select the most important features from the original data as the input of the model. It can help to use fewer resources and less data to solve the problem.

On the basis of the original features, the time difference (s) between adjacent samples is calculated as the new feature variable. Calculate the mileage difference between adjacent samples as the new status feature variable. Due to the large number of features, it is necessary to select the few features that have the greatest impact on the prediction results as the input of the model. This paper first uses Pearson's correlation coefficient to measure the degree of correlation between the mileage difference and other variables. The calculation formula of the correlation coefficient is shown in (10).

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2\sum_{i=1}^{n}(y_i-\bar{y})^2}} \tag{10}$$

Where n is the number of samples, x_i and y_i are the values of the two variables, and $\bar{x}$ and $\bar{y}$ are the mean values of x and y respectively. If |r|>0.5, it is considered to have a strong correlation with mileage; if 0.3<|r|≤0.5, it is considered to be moderately related to mileage; the rest are weakly correlated or have no correlation. **Table. 2** shows the correlation coefficient between "mileage_diff" and other features.

Table. 2 The correlation coefficient between "mileage_diff" and others

feature	**correlation coefficient**
total_voltage	-0.352
total_current	0.187
temp_max	0.083
temp_min	0.079
motor_voltage	-0.28
motor_current	0.293

Refer to **Table .2**(continued)

soc	-0.048
mileage	0.175
hour	0.216
month	0.075

In order to ensure that the feature variables that are not used in the model will not contribute to the improvement of the accuracy, the feature is removed from the input in turn and then train the model in 20 epochs. If the loss function does not increase after the feature is removed, the feature is considered to be useless.

The final selected characteristic variables as input are the following six variables: total voltage, total current, hours of time, motor controller input voltage, motor controller DC bus current and accumulated mileage.

After determining the input variables, the sliding window method is used to construct the data set. Each new sample is composed of continuous samples of a fixed size, which increases the feature quantity of the sample and shows the time sequence of the sample. The target to be predicted is the sum of the mileage difference in the window. In the process of building the dataset, the raw data are first lined up, and the window is sequentially sliding in order from left to right to construct a matrix. If we set the sliding window size to x, the original data size to n, the number of samples in the dataset to m, and we can get m by (11). In this paper x is 50.

$$m = n - x + 1 \tag{11}$$

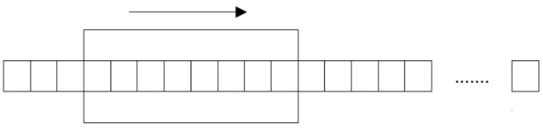


Fig. 5 Sliding Window

Fig. 5 shows that a matrix of size $m \times 50 \times 1$,which is the dataset required for model's training, is obtained by sliding window,

3.3 Data Standardization

In the original data ,the value of the feature "total_voltage" ranges from 500 to 600, while the value of the feature "hours" only ranges from 0 to 24.It is not proper to put them into the network directly because it may cause a bigger update of gradient when training the model, and thus causes the network to fail to converge. So it is necessary to standardize the data before put in the model to speed up the convergence of the neural network. In this paper, deviation standardization is used. It is a method to make the original data fall into [0,1]. The conversion function is shown in (12).

$$x' = \frac{x-min(x)}{max(x)-min(x)} \tag{12}$$

Where $\boldsymbol{x}$ is the original data, $\boldsymbol{min(x)}$ is the minimum value of $\boldsymbol{x}$,and $\boldsymbol{max(x)}$ is the maximum value of $\boldsymbol{x}$.

3.4 Build the Model

The model is mainly consisted of three causal convolution layers and two LSTM layers. The main structure of the model is shown in **Fig. 6**.

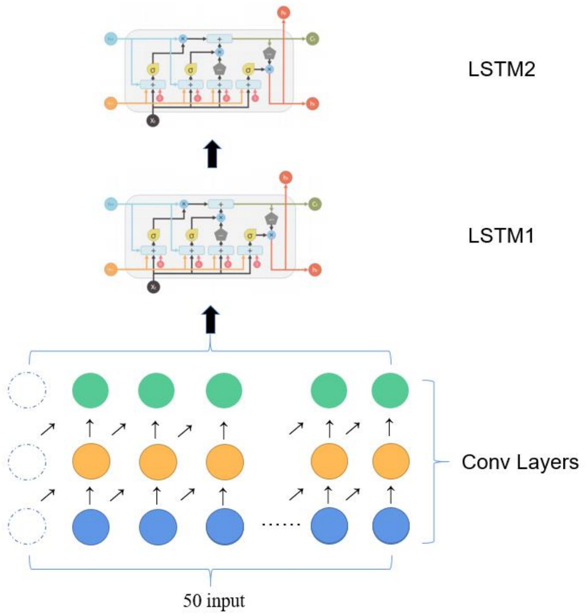
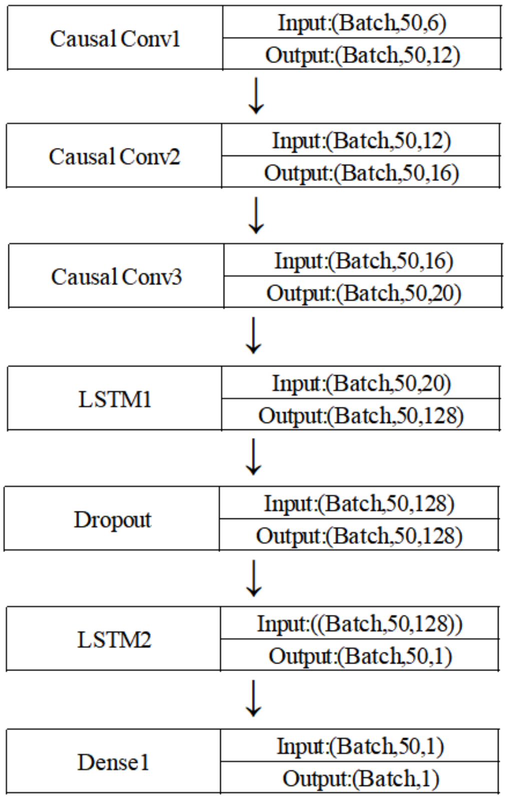


Fig. 6 The structure of Conv-LSTM Model

The structure designed to use the causal convolutional layers to extract data information, and then use the LSTM layers to learning information of time series.

The process of the change of tensors' shape when passing through the network is shown in **Figure. 7**.



Causal Conv1 | Input:(Batch,50,6) | Output:(Batch,50,12)

Causal Conv2 | Input:(Batch,50,12) | Output:(Batch,50,16)

Causal Conv3 | Input:(Batch,50,16) | Output:(Batch,50,20)

LSTM1 | Input:(Batch,50,20) | Output:(Batch,50,128)

Dropout | Input:(Batch,50,128) | Output:(Batch,50,128)

LSTM2 | Input:((Batch,50,128)) | Output:(Batch,50,1)

Dense1 | Input:(Batch,50,1) | Output:(Batch,1)

Fig. 7 The change of tensors' shape

- The model has 7 layers.
- The first ,second and third layers are the causal convolution layers, which are designed to extract the information from the input. The size of the convolution kernel is 2 × 1， and the step is 1.The padding method, which means adding zero at the left of the input, is used to make the dimension of the output match the input's. The numbers of the convolution kernel are 12,16 and 20 respectively.
- The fourth and sixth layers are the LSTM layers, which are designed to learn the information of time series.
- The fifth layer is the dropout layer, which mainly refers to the random update of the weight by a specific probability during the backpropagation process, and other weights temporarily stop updating, mainly used to solve the gradient disappearance and gradient explosion problems.
- The seventh layer is the fully connected layer for outputting the final predicted result

4.EXPERIMENT EVALUATION

An evaluation index is needed to describe the difference between the prediction of the model and the real number. In this paper, RMSPE is used as the evaluation index, and it is shown in (13).

$$e = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{r_i-a_i}{a_i}\right)^2} \tag{13}$$

where r_i is the predicted driving range of the i-th driving segment, a_i is the actual driving range of the i-th driving segment, and n is the total number of driving segments. This indicator is the average value of the error of n segments, which can eliminate the influence of the actual driving range on the evaluation index.

There are a total of 300 car driving segments used to test, and the length of each segment is shown in the **Fig. 8**.

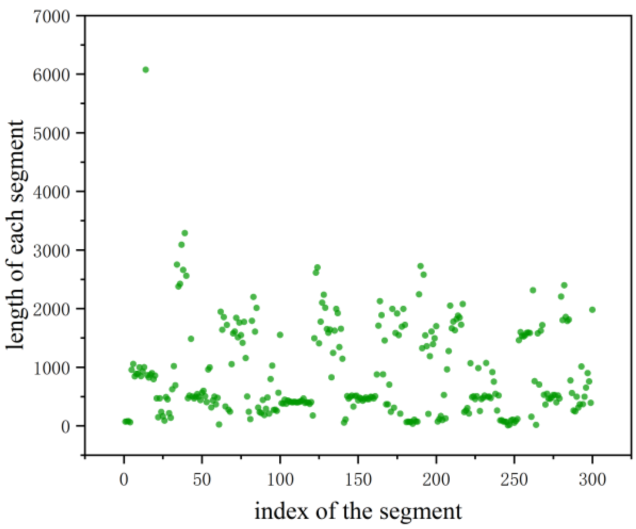


Fig. 8 Length of each segment of the test data

The size of the sliding window used in this paper is 50,so the length of the test data should be a multiple of 50.The data collected then the vehicle was not moving is used to expand the length of each segment to be a multiple of 50.Run the model on each segment of the test data, and calculate the mean value of 300 RMSPE values as the final evaluation.

Fig. 9 shows the RMSPE of the model. It also shows the effect of some methods which are marked beside the points in the figure.

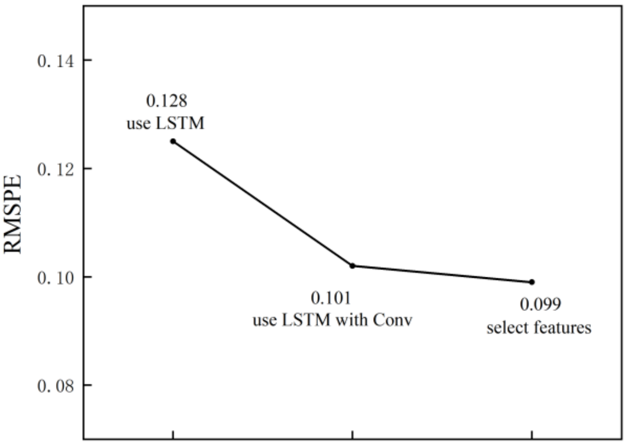


Fig. 9 The evaluation of the model

It can be seen from the figure that the error of the ordinary LSTM model is 0.128. After the causal convolution is applied, while maintain the structure and parameters of the model, the evaluation error drops to 0.101. It means that the convolution layers extract some information from the original input data and generate some useful features.

The selection of the features also do good to the model. It may help reduce the possibility of learning useless information. During the test experiment, some abnormal data such as missing values are also found in the test data. It will help to deal with them first before doing the prediction.

5.CONCLUSION

This paper first cleans the data to deal with abnormal values. Then, feature engineering is applied to extract effective features. After that, the sliding window method is used to construct the datasets, and the data are divided into a training set and a test set. Finally, a novel method using LSTM combined with causal convolution is presented. The final experimental results show that the accuracy of the method is high, and the RMSPE is 0.091. The followings are some useful methods to improve the accuracy of model's prediction.

i. Analyze the correlation of the features, and choose the features that matter most as the input. It may help the model reduce the possibility of learning useless information.

ii. If there are some abnormal data in the test data, deal with them first. This may help to improve the accuracy of the prediction

iii. Using sliding window is a very effective way of expanding the training data, and thus help improve the model.

The followings are some methods can be tried in the future research.

i. Add Attention Mechanism to the LSTM model. It may help the model pay more attention to the most important feature.

ii. Use dilated convolution to replace causal convolution. It can make neurons of convolution layers have bigger receptive field, which means each neuron of the convolution layers' output contain the information that span a wider time range.

REFERENCES:

[1] Cunningham I , Burnham K . Online Use of the Fuzzy Transform in the Estimation of Electric Vehicle Range. Measurement and Control -London- Institute of Measurement and Control-, 2013, 46(9):277-282.

[2] Liu G , Ouyang M , Lu L , et al. A highly accurate predictive-adaptive method for lithium-ion battery remaining discharge energy prediction in electric vehicle applications. Applied Energy, 2015, 149(jul.1):297-314.

[3] F. Kaytez, M. C. Taplamacioglu, E. Cam, F. J. I. J. o. E. P. Hardalac, and E. Systems, "Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines," vol. 67, no. 67, pp. 431-438, 2015.

[4] WANG Z, WANG X, WANG L, et al. Research on electric vehicle (EV) driving range prediction method based on PSO-LSSVM. IEEE, 2017.

[5] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks". 1989.

[6] S. Hochreiter and J. J. N. C. Schmidhuber, "Long Short-Term Memory," vol. 9, no. 8, pp. 1735-1780, 1997.

[7] Hong J, Wang Z, Chen W, et al. Synchronous multi-parameter prediction of battery systems on electric vehicles using long short-term memory networks. Applied Energy, 2019, 254: 113648.

[8] Hong J , Wang Z , Yao Y . Fault prognosis of battery system based on accurate voltage abnormity prognosis using long short-term memory neural networks. Applied Energy, 2019, 251.

[9] Li X , Zhang L , Wang Z , et al. Remaining useful life prediction for lithium-ion batteries based on a hybrid model combining the long short-term memory and Elman neural networks. Journal of Energy Storage, 2019.