A Flow Prediction Model of Bike-Sharing Based on Cycling Context

Yizhu Zhao, Jun Zeng(✉), Min Gao, Wei Zhou, and Junhao Wen

School of Big Data and Software Engineering, Chongqing University, Chongqing, China
{zhaoyizhu,zengjun,gaomin,zhouwei,jhwen}@cqu.edu.cn

Abstract. The prediction of the number of bike-sharing is of great significance to maintain the balance of the number of bikes at each station. The cycling trajectory of users is dynamically changing and different in different districts of a city. This has caused the problem of no bikes at some stations, while others have accumulated bikes. However, most of the research work adds contexts such as spatiotemporal and weather features to the bike flow prediction, but ignores the problem of the imbalance of the number of bikes at each station. Therefore, we predict the number of bikes at the station based on the context features. To this end, we study the context features based on user's cycling data, and consider the features of time and climate. Along this line, we first analyze the features of time and climate to find the user's cycling habits. Then, we introduce the Long Short-Term Memory (LSTM) to capture the dependence relationship between time series. Using the Attention Mechanism to obtain key features can reduce prediction errors. We propose the context-based prediction model of the number of bike-sharing on the station with LSTM and Attention Mechanism (C-LSTMAM). This model can specifically capture more important context feature for the prediction. Finally, extensive experiments on real-world datasets demonstrate the effectiveness of the C-LSTMAM.

Keywords: Prediction of the number of bike-sharing · Context analysis · Long Short-Term Memory · Attention mechanism

1 Introduction

With the construction of Smart City and the increment of people's trips, bike-sharing is widely used in cities [1, 2]. According to Statista's estimation, as of May 2018, more than 1,600 bike-sharing programs were in operation worldwide, providing more than 18 million bikes for public use. With the increase of the number of bikes and users, bike-sharing companies can collect a great scale of cycling information [3, 4], which includes the user's cycling time, starting place, single cycling time and whether to buy a cycling card. However, because everyone's travel path is different, there will be a phenomenon of stacking bikes at some stations [5]. This phenomenon makes the use of bike-sharing become very unbalanced, leading to the situation that some stations have no bikes, while others have abundant bikes [6]. Therefore, it is necessary to predict the

H. Gao et al. (Eds.): CollaborateCom 2022, LNICST 460, pp. 73–92, 2022.
https://doi.org/10.1007/978-3-031-24383-7_5

number of bike-sharing on the station to balance the number of bikes at each station in real life.

The existing bike-sharing systems have comprehensive functions [7], but it is still challenging to effectively predict the number of bike-sharing each station so as to balance the bike demand of each station. Researchers use data mining related technologies to analyze user behavior from user historical cycling data [4, 8]. According to the user's cycling trajectory predict the number of bike on the station, so as to assist bike managers to allocate bikes reasonably [3, 9]. The number of bike-sharing prediction problem of bike-sharing is to predict the number of bike-sharing at each bike station in the future, which is based on the user's historical cycling data. Many scholars have solved the problem [5, 6, 10, 11], but the prediction accuracy still needs to be improved. Due to the mature development of deep learning and its good feature extraction ability and robustness, researchers generally use deep learning to study the flow prediction problem of bike-sharing [12, 13]. Although machine learning or neural network models can capture the features of time series, they often lack the ability to extract features dynamically. Therefore, some researchers [14–16] propose that the flow association pattern which can be abstracted into a graph structure. Chai et al. [15] expresses the bike-sharing system with a weighted graph, where the nodes are the stations, the edges are the relationships between the stations, and the weights of the edges represent the strength of the relationship between the stations. This method ignores the factors of the riding environment. Deng et al. [17] use Convolutional Neural Network (CNN) to extract temporal and spatial features of the grid, and model the time dependence between any two regions. The construction of dynamic time series models to extract effective features plays the important role in the prediction of the number of bikes.

In prediction problem of the number of bike-sharing, the user's cycling at any time always be affected by the previous moment, and there is a strong correlation between these effects. The Long Short-Term Memory (LSTM) can deal with long sequences of data and time series processing. What's more, LSTM can mine these connections, including information about the current node and important information at the previous moment. However, the results of each step of LSTM are dependent on the results of the previous step, so parallel computing is not possible. There is no result dependence in Attention Mechanism, so it can be processed in parallel and retain the previous information of LSTM. In addition, Attention Mechanism can grasp the key content of text or other information and assign different weights according to the importance of the information, so as to obtain more effective information. We introduce the Attention Mechanism to LSTM, which can help the model to retain the connection between time series, and it can also capture the more important context feature information for the prediction moment.

We analyze the features of the context information of users' historical cycling data, including the features of time and climate. The LSTM is used to capture the dependence between time series, and combine the Attention Mechanism to build a dynamic the number of bike-sharing prediction model. By analyzing the time context features of the datasets, it is found that the fluctuation of the data presents certain regularities. These regularities are called trend, periodicity, and proximity according to their internal relationships. Further analysis of these regularities, we found that weekends and

weekdays, morning and evening rush hours have different effects on cycling demand. Therefore, when predicting the number of bike-sharing, we construct weekends and weekdays, morning and evening rush hours as new features to assist the model to learn users' cycling habits, and further improve the prediction accuracy. We also statistical analyze the climate context feature of the datasets such as weather, wind speed, pressure, temperature and humidity. The analysis results show that weather, wind speed, pressure and temperature have an effect on users' riding times. Besides, the linear relationship between humidity and user riding times is irregular. After analyzing the average filtering of the data, the linear relationship between humidity and user riding times is still irregular, so we regard humidity as redundant features.

The main contributions of this paper are summarized as follows:

- In order to balance the number of bikes at each station, we fully explore features such as climate and user riding habits to predict the number of bikes parked at station in the future. In context analysis, we extract the features of time and climate, and exclude the redundant feature humidity.
- In order to predict the number of bikes at station, the main work of the research is to mine the relationship between these features and the number of rides by users based on the time context and climate context.
- Compared with the problem of bike flow prediction, we are more concerned about balancing the number of bikes at each station in the future by predicting the number of bikes, so as to solve the situation of users without bikes at station.

The rest of the paper is organized as follows. Section 2 summarizes the related work, which is highly relevant to the research. Section 3 describes the analysis of context features. Section 4 provides detailed methodology of we proposed model. Section 5 presents experiments and the results, and Sect. 6 concludes this paper and outlines prospects for future study.

2 Related Work

Bike-sharing has become a necessary transportation tool for urban residents. The huge users produce hundreds of millions of behavioral data, and the value hidden behind the data has attracted wide attention from both academia and industry [18–21]. Lihua et al. [22] make prediction based on the features of non-linearity and different time and space in the cycling data, and used the good linear fitting ability of the Auto-regressive Integrated Moving Average model to process the data. However, ARIMA model can only consider the features of time level, which makes the prediction ability relatively weak. To solve this problem, Zhang et al. [23] propose a hybrid model based on Seasonal Auto-regressive Integrated Moving Average model (SARIMA) and Support Vector Machine (SVM) model by using the periodicity, non-linearity, uncertainty and complexity of short-term traffic flow prediction to predict time series. SARIMA model can find the correlation between time series, especially suitable for the modeling of seasonal and random time series. SVM has strong nonlinear mapping ability for input and output data. They mixed the two models and combined their advantages. Compared with the traditional ARIMA

model, this model takes into account the influence of different seasons on the flow prediction, and the accuracy has been greatly improved.

With the development of machine learning, researchers have gradually weakened the use of time series modeling methods. Ahn et al. [24] propose a real-time flow prediction method based on Bayesian Classifier and Support Vector Regression (SVR). They use 3D Markov to model the flow of road traffic and its relationship in time and space, and divide the regions with close relationship together. Multiple Linear Regression and SVR are used to estimate the dependence between regions, so as to predict the traffic flow. Although the relationship between regions is considered in this method, the correlation of traffic flow between different roads in the same region is not considered. Traditional machine learning methods generally focus on the modeling of time and space, which are two kinds of features in the dataset. However, without other features, the accuracy is limited.

Compared with traditional machine learning, deep learning is favored by researchers for its ability to solve complex problems. Lv et al. [12] not only consider the temporal and spatial features of traffic flow prediction involved in traditional methods, but also used stacked Auto-Encoder to reduce the dimension of the data so as to complete the feature extraction. Finally, the output of the last layer of the Auto-Encoder is taken as the input of a regression network for supervised learning to complete flow prediction. Compared with the traditional machine learning, the prediction accuracy is improved. But like the traditional methods, it only considers the temporal features covered in the data, and does not do additional feature engineering. Zhang et al. [13] divide the urban area into large and small grids, and used Convolutional Neural Network (CNN) to extract temporal and spatial features in the grid. However, the author did not conduct a comprehensive and detailed analysis of the dataset or select the features, resulting in the problem of feature redundancy, which affected the final prediction results. Besides, users' travel rules and cycling preferences change over time. Machine learning or neural network models can capture temporal features, but they often lack the ability to extract features dynamically. The Attention Mechanism can grasp the feature factors corresponding to each moment. Therefore, the introduction of Attention Mechanism on the basis of sufficient feature engineering can construct dynamic time series model. The more important features are captured from the historical data, while the unimportant features are selectively ignored. Therefore, we focus on analyzing the historical data of user's cycling habits to find out the main features that affect the flow prediction of bike-sharing. In a word, we propose a prediction of the number of bikes model based on LSTM and Attention Mechanism.

3 Analysis of Context Features

In this section, we conduct contexts analysis of users' cycling data. The user's riding data includes the riding records of the bike and the climate features of the day of riding. According to experience in life, users choose bike-sharing as a way to travel in a suitable climate. The suitable or bad weather affects users' cycling behaviors, which also affects users' demands for bike-sharing. Therefore, we analyze the time context and climate context respectively to study their influences on users' cycling behavior. The datasets

include cycling records of Citi Bike [25] and climate data obtained from the website of Weather Spark[1].

3.1 Analysis of Time Context

We analyze the period of the dataset in March 2017 by hour, which consist of 15 features, 114,698 rows of data and 619 bike stations. The analysis result is shown in Fig. 1. The tendency of a broken line to move up or down over a continuous period of time is called trend. A period of one week is called periodicity when the same period has a similar trend in the direction of the broken line. On the other hand, a period of one day is called proximity.

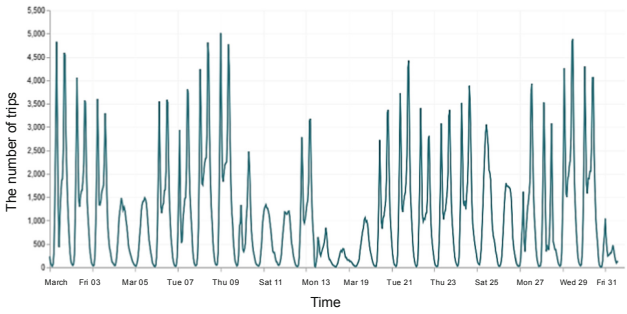


Fig. 1. The trend chart of the number of trips with time

We conduct further analysis on the datasets. The data for March 2017 was aggregated in hours, and the total number of rides by users in March was counted. According to hourly clustering, the periodic features of the number of bikes on the site can be counted, which are the number of bikes that we predict at a certain moment in the previous hour, at the same time in the previous day, and at the same time in the previous week. Besides, the weather data is based on hourly statistics, and the hourly data is different. Therefore, the user's riding habits features can be analyzed, morning peak and evening peak from the context of time and climate. The data distribution of the top 5 stations in the total number of rides is shown in Fig. 2. There are two obvious peaks, which are 7 am to 9 am and 5 pm to 6 pm. The phenomenon is consistent with the user's daily routine, the commuting time, which is called rush hours. Based on this data analysis, the main users of bike-sharing may be commuters. Commuters' travel has obvious regularity, which can be used to construct features advantageously. In order to verify this conclusion, we use Baidu Map API to analyze users' cycling heat map. Figure 3 (a) shows the heat map in the morning rush hour, and Fig. 3 (b) shows the heat map in the evening rush hour. After converting the number of rides to space, commuters mostly work in the city center, while they often live around the city. At different times, users have different travel patterns and demand for bike-sharing, which fully shows that it is correct to conclude that the main users of bike-sharing are commuters.

[1] https://zh.weatherspark.com/.

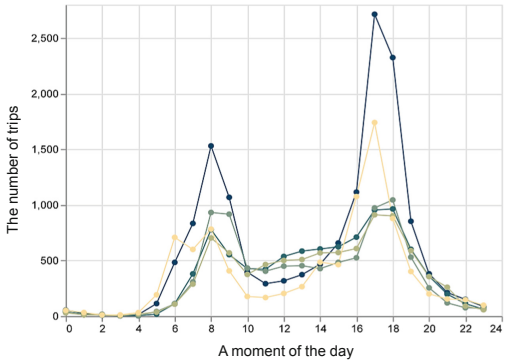


Fig. 2. The ridership of the top 5 stations in one day

(a) The morning rush hour (b) The evening rush hour

Fig. 3. The heat map of the top 5 stations in one day

According to experience in life, commuters' demands for bike-sharing vary in time. Therefore, we analyze the dataset based on weekends and weekdays, as shown in Fig. 4. The number of trips at each station varies greatly on weekends and weekdays. This data distribution is consistent with experience in life, which is that commuters go to

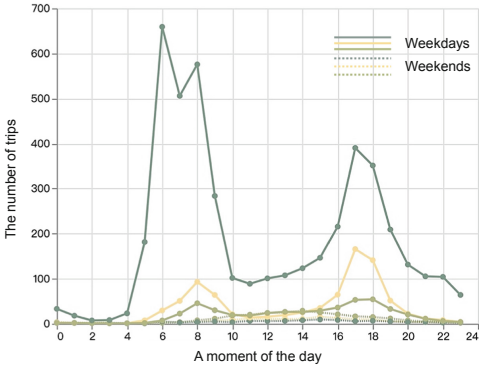


Fig. 4. The trips number of top 5 stations on weekdays and weekends

work from Monday to Friday, and the demand for bike-sharing peaks in the morning and evening from Monday to Friday. However, commuters take a rest on weekends, and there is no obvious peak on weekends. Based on these data analysis, when predicting the number of bike-sharing, we construct new features on weekends and weekdays, as well as morning and evening rush hours. This can help the model learn users' cycling habits and further improve the prediction accuracy of the model.

3.2 Analysis of Climate Context

We crawled the weather data corresponding to the user's riding time from the Weather Spark website. The data contains weather data for New York from June 1st to September 30th, 2017. The weather data has 16 types of weather, which we map into 6 types of weather.

The influence of different weather on cycling demand is shown in Fig. 5, and it shows that users have a great demand for bike-sharing in sunny day. In addition, we also analyze the two features of wind speed and pressure. Figure 6 shows that the wind speed is most suitable for cycling when the wind speed is level 1 to level 4. With the increase of wind speed after Level 4, the cycling demand decreases to varying degrees. Figure 7 shows a great difference in the impact of 1014 kPa and 1015 kPa on the user's cycling demand, even though the pressure difference is only 1 kPa. Therefore, the two features of wind speed and pressure is added to the feature analysis of bike-sharing number prediction. The relationship between the temperature and the number of trips by the user is shown in the Fig. 8. When the temperature is between 27 and 32 °C, the number of rides is the largest, and when the temperature is lower than 25 °C, the number of rides is lower. Among them, when the temperature is 23 °C, there are more riding times, because in addition to weather features, there are other factors such as pressure and wind speed.

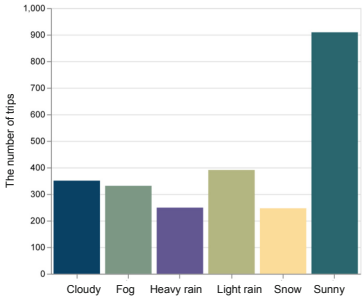


Fig. 5. The influence of weather on the trips number

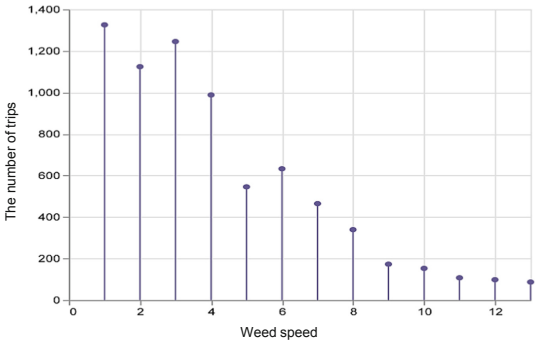


Fig. 6. The influence of wind speed on the trips number

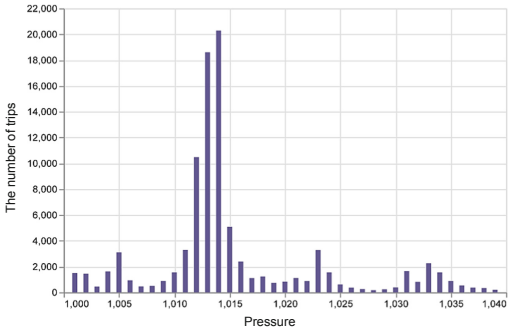


Fig. 7. The influence of pressure on the trips number

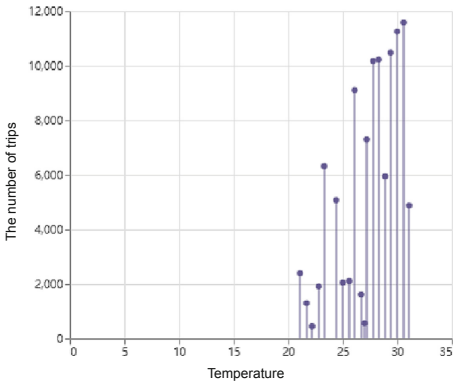


Fig. 8. The ridership of the top 5 stations in one day

We also analyze the feature of humidity, but the influence of humidity on the number of trips is not regular, and the results are shown in Fig. 9. Humidity is directly affected by weather, so the humidity is considered as redundant features. In order to verify the rationality of removing redundant features, we analyze the influence of features on the

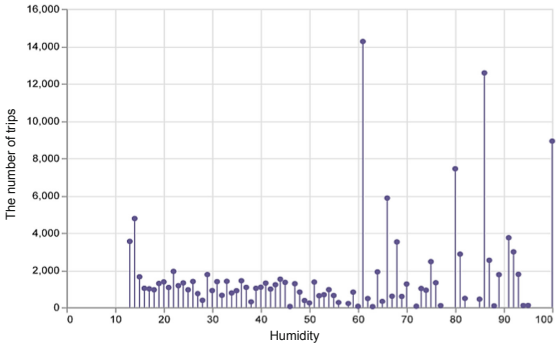
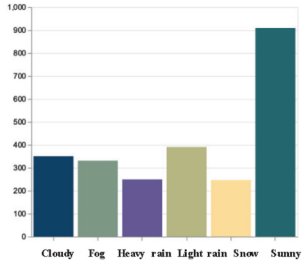
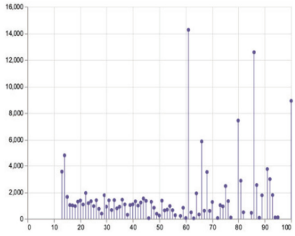


Fig. 9. The influence of humidity on the trips number

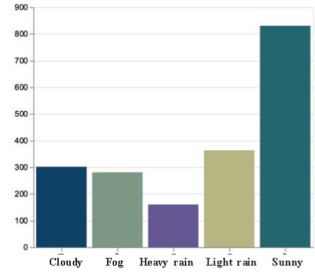
number of rides, namely weather and humidity. This experiment is to compare the effects of weather and humidity on ridership by filtering out the data that are lower than the average the number of trips.



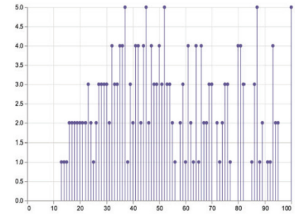
(a) Weather (before)



(b) Humidity (before)



(a) Weather (after)



(b) Humidity (after)

Fig. 10. The influence of weather and humidity on the trips number

In the dataset with humidity features, the average number of trips in the user's historical cycling data is 3602, and the data is filtered based on the average number. Figure 10 shows the influence of weather and humidity on the cycling demand of bike-sharing. Comparing Fig. 10 (a) with Fig. 10 (c), the snow type of weather disappeared

after filtering the data, which shows that bad weather has a negative impact on the cycling demand of bike-sharing. By comparing Fig. 10 (b) with Fig. 10 (d), the influence of humidity on the cycling demand of bike sharing is still irregular after filtering the data. Humidity is directly determined by the weather, therefore, the humidity as redundant features. Deleting the redundant feature of humidity when constructing features of the model can further improve the accuracy of model prediction.

4 Framework

In this section, first, the definition of the problem is presented. Then, the details of the context-based prediction of the number of bike-sharing model with LSTM and Attention Mechanism (C-LSTMAM) is introduced. The Long Short-Term Memory (LSTM) is used to capture the dependence between time series, and a dynamic prediction model is constructed by combining Attention Mechanism. The model structure is shown in Fig. 11.

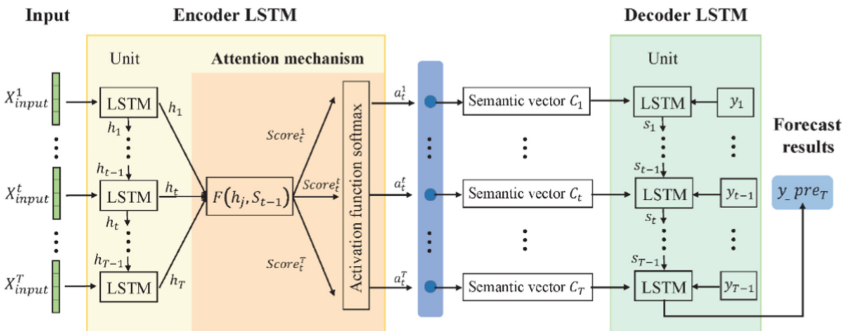


Fig. 11. The model structure diagram

4.1 Problem Definition

First, the notations of this paper are shown in Table 1. Next, the definitions are described.

For any station, x_t is called the observation value corresponding to time t in this paper. The number of bike prediction problem includes the historical observation sequence $X = \{x_1, x_2, \cdots, x_T\}$, the target flow sequence $Y = \{y_1, y_2, \cdots, y_{T-1}\}$ and the predicted value $\widetilde{y_T}$, where T is the time step size. The number of bike prediction is to use the observed data of the first T hours to predict the flow value of the next one hour. y_1 represents the station flow corresponding to the historical observation sequence value x_1.

According to the previous analysis, all historical observations are divided into three parts:S_{hour}, S_{day}, S_{week}. The difference between the three parts lies in the different feature window w. When the feature window w is 1 h, and the time step T = 3, the unit is w. T is the length of sampling to obtain historical data as feature. When the value of w is 24 h

Table 1. The notations of this paper.

Notation	Definition	Notation	Definition
x_t	The observed the number of bike-sharing value at time T	y_t	The target the number of bike-sharing value at time T
X	Historical the number of bike-sharing observation sequence	Y	Target the number of bike-sharing sequence
T	Time step	w	The feature window
S	The historical observations	E	The context features
h_t	The hidden layer state in the encoder at time T	s_t	The hidden layer state in the decoder at time T
b	The offset vector	U	The weight matrices
V	The weight matrices	W	The weight matrices
$\widetilde{y_T}$	The predicted the number of bike-sharing value at time T	C_t	The semantic vector at time T
a_t^i	The attention weight of the i-th input at time T	Score_t^i	The attention score of the i-th input at time T
X_t^i	The corresponding input features at time T		

and T = 3, it means that the historical the number bike of three days before the forecast period is used as feature. When the value of w is 148 h and T = 3, it means that the historical bike of three weeks before the forecast period is used as feature. Equation (1) is expressed as follows:

$$\mathrm{S} = \left[X_{t-Tw}, X_{t-(T_w)}, \cdots, X_{t-1}\right] \tag{1}$$

When w is 1 h, 24 h and 148 h, S is recorded as $S_{hour}, S_{day}, S_{week}$. The context features introduced in this paper include time, weather, weekends and weekdays. The forecast period refers to the time period of the day or the day of the week. The context feature is represented by E, as shown in Eq. (2).

$$\mathrm{E} = \{\text{weather, windspeed, pressure, dayofweek, hourofday}\} \tag{2}$$

Then the input feature corresponding to time t is $X_{input}^t = \{S_{hour}^t, S_{day}^t, S_{week}^t, E_t\}$. Finally, the prediction problem of bike-sharing is defined as: input all X and part of the observation sequence $\{y_1, y_2, \cdots, y_{T-1}\}$ within the time step T, and calculate the number of bikes $\widetilde{y_t}$ corresponding to the time T, as shown in Eq. (3).

$$\widetilde{y_T} = F\left(X_{input}^1, X_{input}^2 \ldots X_{input}^T, y_1, y_2 \ldots y_{T-1}\right) \tag{3}$$

where E_t is the context feature corresponding to time t. F is a nonlinear mapping relation to be learned.

4.2 Feature Extraction of Time Series Based on LSTM

The essence of the number of bike-sharing prediction is to deal with the problem of seq2seq. The seq2seq problem refers to that both input and output are sequences, such as machine translation [26], question answering system, document abstract, etc. The Encoder- Decoder model [27] is suitable for the seq2seq sequence problem. In the study of the number of bike-sharing prediction, we use the encoder to convert the input historical sequence and corresponding features into a fixed-length semantic vector C, and the decoder is responsible for converting the semantic vector C generated by the encoder into the output of prediction results, as shown in Fig. 13.

We use the LSTM [28] model based on the framework of encoder and decoder, which uses the classic three-layer gate structure for each neuron, namely forget gate, input gate and output gate. LSTM retains the advantages of Recurrent Neural Network in time series processing, and the structure of gate can delete or add information to the cell, which overcomes the problem that RNN cannot capture the long term dependence. The input of encoder is $\{X^1_{input}, X^2_{input} \ldots X^T_{input}\}$, in LSTM, the hidden state of current time t is determined by the state h_{t-1} of the previous time and the input x_t of the current time. Then according to Eq. (4), the hidden state in the encoder can be obtained.

$$h_t = f_1(h_{t-1}, x_t) \tag{4}$$

The encoder reads the input data of historical $\{X^1_{input}, X^2_{input} \ldots X^T_{input}\}$ one by one and encodes them as a hidden state sequence $\{h_1, h_2 \ldots h_T\}$. These hidden states contain the feature of the original input data. The semantic vector C is formed by the rule of q, which is used by the decoder. A simple method is to generate semantic vector C by obtaining the final hidden layer directly, as shown in Eq. (5). However, the feature of the T−1 moments of last time is ignored. The final prediction result is only related to the state of hidden layer at the last moment, which leads to large prediction error.

$$C = q(\{h_1, h_2 \ldots h_T\}) = h_T \tag{5}$$

Decoder can be regarded as the inverse process of encoder. The decoder is used to combine the semantic vector C and the part of the observation sequence $\{y_1, y_2 \ldots y_{T-1}\}$ to predict the next output value $\widetilde{y_T}$. The LSTM is still used in the decoder section, and The formula for predicting $\widetilde{y_T}$ is shown in Eq. (6).

$$\widetilde{y_T} = g(s_t, y_{T-1}, C) \tag{6}$$

where s_t is the state of hidden layer corresponding to time t in LSTM. The semantic vector C is the output of the encoder, which contains the input information after encoding. y_{T-1} is the output of time T−1, and it's also the input at time T. g is LSTM in the decoder.

Although the Encoder-Decoder model is classical and can solve the seq2seq problem well, it also has certain limitations. Because the only connection between encoder and decoder is a fixed length semantic vector C, which requires the encoder compresses the entire input data of historical sequence into a fixed-length vector. There are two disadvantages. First, the length of the semantic vector C is limited. In other words, only part of the features is encoded and the semantic vector C cannot completely represent

the information of the whole sequence. Secondly, the information in the front of the time node will be diluted or even overwritten by the information in the back. The input sequence is sometimes very long in the number of bike-sharing prediction problem, and the semantic vector C cannot obtain a lot of useful information from the input sequence due to the limitation of the encoder. Therefore, the accuracy will be reduced when decoding. In order to solve this problem, attention mechanism is added to the encoder, which can retain the features more related to the prediction results while ignoring the relatively unimportant features.

4.3 Computation of Important Features Based on Attention Mechanism

Attention mechanism [29] is a technology that can make the model focus on important information and fully learn. In the prediction of bike-sharing, $\{X^1_{input}, X^2_{input} \ldots X^T_{input}\}$ are taken as the input, which are not all important to the result of the moment t. Some of the input sequences of features have great influence on the prediction results, while others have little influence.

The traditional LSTM model assigns the same weight to all input features. In the problem of the number of bike-sharing prediction, if predict the number of bike-sharing at 6 pm, the data at 5 pm is more important for the prediction at 6 pm, but the data at 3 pm is relatively weak effect on the results. In order to distinguish the importance, the semantic vector C in the encoder can obtain the feature information of the input sequence more effectively and completely, and we introduced the Attention Mechanism into the encoder. We use LSTM model to encode the input time series $\{X^1_{input}, X^2_{input} \ldots X^T_{input}\}$ to the hidden layer state corresponding to each input feature, and accumulate the hidden vector sequence $\{h_1, h_2 \ldots h_T\}$ by weighting, as shown in Eq. (7).

$$C_t = \sum_{i=1}^{T} a_t^i h_i \tag{7}$$

Then, the encoder encode the input information into a semantic vector sequence $\{C_1, C_2 \ldots C_T\}$, which contains feature information that is more important to the prediction results at the corresponding time. The LSTM model is used when encoding, where h_i contains the i-th input sequence and some of the previous sequence information in the input sequence of features. The hidden layer vectors are added according to the weights, which means that the attention distribution is different when the output at time t is generated. The larger the value of a_t^i, the more attention is allocated to the output corresponding to the time t on the $i - th$ input sequence. And a_t^i is jointly determined by the corresponding output hidden state s_{t-1} at time t−1 the hidden layer states in the input, as shown in Eq. (9). In order to distinguish the state of hidden layer between encoder and decoder, the state of the hidden layer in the encoder is h_t and the state of the hidden layer in the decoder is s_t at time t. *tanh* is activation function.

$$Score_t^i = V^T tanh(W[h_i, s_{t-1}]) \tag{8}$$

$$a_t^i = softmax(Score_t^i) = \exp(Score_t^i) / \sum_{j=1}^{T} \exp(Score_t^j) \tag{9}$$

The above equation represents a nonlinear mapping relation, which can make s_{t-1} and the hidden layer state h_i corresponding to the input vectors of feature calculate to get a value, and then use softmax to get the attention weight at time t. Each influence factor is given a certain weight to represent the importance of the input features. During decoding, the corresponding semantic vector C_t is used for decoding. C_t contains the most important part of the information and ignores the unimportant feature, which makes the prediction errors more less.

4.4 The Model Based on LSTM and Attention Mechanism

As discussed above, the Encoder-Decoder model can deal with the seq2seq problem such as the number of bike-sharing prediction. Due to the limitation of encoder, semantic vector C cannot obtain enough useful information of input sequence. The Attention Mechanism enables the model to focus on important information and fully learn, which is no longer limited to encode all input information $\{X^1_{input}, X^2_{input} \ldots X^T_{input}\}$ into a fixed length semantic vector C, but to encode the input information into semantic vector sequence $\{C_1, C_2 \ldots C_T\}$. Each semantic vector contains more important feature information for the results of prediction at corresponding time, which makes up for the deficiency of Encoder-Decoder model, LSTM decodes the semantic vector sequence to get the final prediction result is shown in Eq. (10).

$$s_t = f_2\big(s_{t-1}, U[y_{t-1}; C_{t-1}\big] + b) \tag{10}$$

where y_{t-1} is the observed value corresponding to time t−1, which is the number of bike-sharing prediction corresponding to time t−1. C_{t-1} is the semantic vector corresponding to time t−1, which contains the input feature information most relevant to the value of prediction at time t-1. $\big[y_{t-1}; C_{t-1}\big]$ is to connect the two algorithms and use them as the input of LSTM network together with the hidden layer state corresponding to time t−1. f_2 is calculated by the LSTM model, and U and b are the learning parameter in the network.

The final results of prediction can be obtained by using Eq. (11). Where $\widetilde{y_T}$ is the value of prediction corresponding to time T. V, W, b_w and b_v are all parameters that need to be learned in the network. s_T is the hidden layer state of decoder at time T, and C_T is the semantic vector obtained by encoding corresponding to time T.

$$\widetilde{y_T} = V(W[s_T; C_T] + b_w) + b_v \tag{11}$$

5 Experiments

In this section, the effectiveness of proposed method is demonstrated by utilizing the datasets of Citi Bike. We compare model with other baseline methods, analyze the generalization of our approach, and evaluate the accuracy of the model prediction.

5.1 Datasets

Citi Bike [25] has collected user history ride data since 2013. Following other researcher, we use 114,698 rows of data from June 1st to September 30th, 2017. The climate data of the same period is added, and the details of the final experimental dataset including the context features are shown in Table 2. In the experiment, 70% of the total data is selected as the training set and the remaining 30% as the test set.

Table 2. The details of the dataset.

Details of dataset	Information	Details of dataset	Information
Place	New York	Number of Stations	621
Time Span	2017/06/01–2017/09/30	Weather	6 types (light rain, snow…)
Data Field	15 species	Temperature	[31.1, 21.1]
Missing Value	Age (missing rate 0.5%, mean filling)	Wind Speed	[0, 12]
Number of Data	114,698	Pressure	[995, 1040]

5.2 Setting

In our method, we use encoder and decoder base on attention and LSTM to predict the number of bikes on the station. We use the deep learning framework Pytorch to perform experiments on NVIDIA GeForce GTX 1650 (with 12G RAM). The model is trained by using the Adam optimizer with a learning rate of 0.001. The batch size is 128 and the dimension of encoder and decoder is 128. The parameters of the baselines are the default values. In experiment, loss gradually decreases with epoch. When the epoch is 47, the model loss is the smallest. Therefore, the epoch is set to 50 and take the parameters when the model is optimal. In the process of data processing, we count the number of bikes used by all users of station in hour. The number of times the bike is used is counted by the starting station when the user uses the car, divided by hourly clusters and stations. Whether it is a weekend or not, it is divided according to the fact that Monday to Friday is set to 0 for the week, and Saturday to Sunday is set to 1 for the weekend. Similarly, whether the peak period is divided into 1 and 0 according to the above time period analysis, 1 refers to the peak period, and 0 refers to the off-peak period.

5.3 Evaluation Metrics

In this paper, Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) are used as the evaluation metric. The Mean Absolute Error can reflect the fitting effect of the model, as shown in Eq. (12). MAE accumulates the error between each predicted value and the real value. The larger the MAE, the greater the prediction error. So a model

with good predictive power should ensure that the MAE is as small as possible. However, RMSE can magnify the value with large prediction error and compare the robustness of different models. As shown in Eq. (13). Where n is the number of test samples, y_i is the true value, and $\widetilde{y_i}$ is the predicted value. In the same way, a model with good predictive power should ensure that the RMSE is as small as possible.

$$MAE = 1/n \sum_{i=1}^{n} |y_i - \widetilde{y_i}| \tag{12}$$

$$RMSE = \sqrt{1/n \sum_{i=1}^{n} (y_i - \widetilde{y_i})^2} \tag{13}$$

5.4 Results

Analysis of Prediction Results

The downtown station has a greater demand for bikes than the suburban station. It can be seen from the Fig. 12 that the model can fit the trend of the number of bikes at each station over time. Figure 12 shows the number of bikes predicted to have a site ID of 223 in the next week. The value of the number of bikes predicted by the model is close to the true value and fits the trend of the true value. This shows the effectiveness of the model for the number of bikes prediction. Similarly, the model can only fit the trend of the true value very well, and cannot accurately predict the specific number of bikes. This is because the number of bikes is related to other features such as geographic and location features. C-LSTMAM cannot predict the maximum peak value because the maximum peak value is an abnormal value in a continuous period of time.

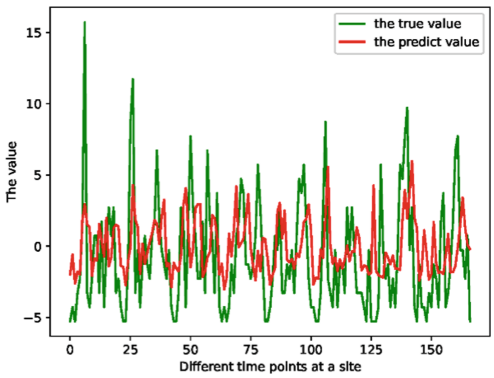


Fig. 12. Predict the number of bikes with a station ID of 223 in the next week (ID = 223)

Comparison to Baselines

We compare the performances of the proposed method against the following baseline algorithms.

RNN [30]: Recurrent Neural Network is used to capture the correlation between time series data to predict future values.

LSTM [31]: The recurrent neural network with gating mechanism is used to mine the long-term dependence of sequence data to predict future values.

XGBoost [32]: XGBoost has good learning effect and fast training speed. It is a machine learning library that focuses on gradient boosting algorithms.

Figure 13 shows the performance results of our proposed C-LSTMAM as compared to all the baselines. MAE and RMSE are all the smaller the equivalent value, the better the performance of the model. Although the recurrent neural network RNN and LSTM are effective in processing time series data, they have poor performance compared to the LSTM model with attention. This is because the attention mechanism can assign different weights to each feature according to the relationship between the data feature and the result, and strengthen the importance of certain features. Compared with RNN, LSTM adds a gating mechanism to mine the long-term dependence between sequence data, so the results on the three evaluation indicators are better. XGBoost is an optimized distributed gradient boosting library that implements decision tree boosting in parallel. It has a stronger performance than LSTM. Similarly, because of the lack of attention mechanism distribution and the inability to mine important features, the performance is lower than the model based on combination of LSTM and Attention. Considering MAE and RMSE indicators, the C-LSTMAM is better than others.

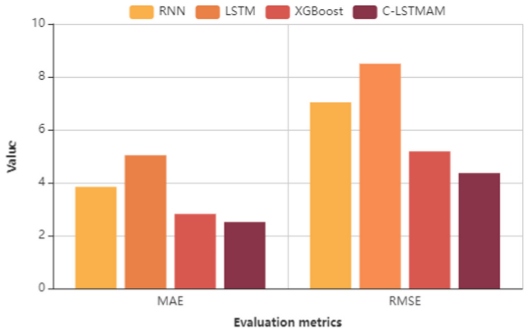


Fig. 13. The comparison with baselines

Context Feature Verification

In order to verify the feature importance analysis, we select the temperature in the feature for an ablation experiment. C-LSTMAM + temp considers the temperature features, while C-LSTMAM-temp removes the temperature features. The results are shown in the Table 3, and the performance of the model taking into account the temperature features is obviously better. This shows that the temperature feature has an impact on predicting the number of bikes on the station, which verifies the importance of considering the riding context.

Table 3. Temperature importance analysis

Methods	MAE	RMSE
C-LSTMAM+temp	2.516	4.368
C-LSTMAM-temp	2.958	4.880

6 Conclusion

In this paper, we analyze the historical cycling dataset of users. Then, select and construct the relevant features. The number of bike-sharing prediction method based on LSTM and Attention mechanism was proposed. The algorithm proposed in this paper not only considers the temporal features, but also introduces the context features of climate related to cycling data. Through the combination of LSTM and Attention Mechanism, the dynamic to predict the number of bike-sharing model is constructed. In other words, this model can extract features dynamically. The proposed model is verified by using Citi Bike dataset, and the experimental results show that the model can reduce prediction errors well. The performance of MAE and RMSE shows the effectiveness of the proposed algorithm, and the rationality of feature selection is verified by experiments. The C-LSTMAM model can predict the number of bikes at station, thereby balancing the number of bikes at station. For future work, we plan to use the K-Means clustering algorithm based on the transition matrix to divide the station into different areas according to the predicted number of bikes. The scheduling optimization is performed by considering the local maximum efficiency, and the Q-Learning method of reinforcement learning is used to schedule bikes between stations in each area.

Acknowledgement. Funding: This work was supported by the Natural Science Foundation of Chongqing, China [No. Cstc2020jcyj-msxmX0900]; and the Fundamental Research Funds for the Central Universities [Project No. 2020CDJ-LHZZ-040].

References

1. Yang, X.-H., et al.: The impact of a public bicycle-sharing system on urban public transport networks. Transp. Res. Part Policy Pract.**107**, 246–256 (2018)
2. Jiang, W., Luo, J.: Graph neural network for traffic forecasting: a survey. 117921 (2022)
3. Chemla, D., Meunier, F., Calvo, R.W.: Bike sharing systems: solving the static rebalancing problem. Disc. Optim. **10**(2), 120–146 (2013)
4. O'Mahony, E., Shmoys, D.B.: Data analysis and optimization for (citi) bike sharing. In: Twenty-Ninth AAAI Conference on Artificial Intelligence, Citeseer (2015)
5. Yang, Z., Hu, J., Shu, Y., Cheng, P., Chen, J., Moscibroda, T.: Mobility modeling and prediction in bike-sharing systems. In: International Conference on Mobile Systems, Applications, and Services (2016)
6. Y. Tang, H. Pan, and Y. J. T. R. P. Fei, "Research on Users' Frequency of Ride in Shanghai Minhang Bike-sharing System," vol. 25, pp. 4983–4991, 2017
7. Wang, B., Vu, H.L., Kim, I., Cai, C.: Short-term traffic flow prediction in bike-sharing networks. J. Transp. Syst. **26**(4), 461–475 (2022)

8. Chang, X., Feng, Z., Wu, J., Sun, H., Wang, G., Bao, X.: Understanding and predicting the short-term passenger flow of station-free shared bikes: a spatiotemporal deep learning approach. IEEE Intell. Transp. Syst. Mag. **14**(4), 73–85 (2021)
9. Hua, M, Chen, X., Chen, J., Jiang, Y.: Minimizing fleet size and improving vehicle allocation of shared mobility under future uncertainty: a case study of bike sharing. J. Clean. Prod. **370**, 133434 (2022)
10. Wang, B., Kim, I.: Short-term prediction for bike-sharing service using machine learning. Transp. Res. Proc. **34**, 171–178 (2018)
11. Singhvi, D., et al.: Predicting bike usage for New York City's bike sharing system. In: National Conference on Artificial Intelligence (2015)
12. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.-Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Trans. Intell. Transp. Syst. **16**(2), 865–873 (2014)
13. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction (2016)
14. Xie, M., Yin, H., Wang, H., Xu, F., Chen, W., Wang, S.: Learning graph-based poi embedding for location-based recommendation. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, pp. 15–24 (2016)
15. Chai, D., Wang, L., Yang, Q.: Bike flow prediction with multi-graph convolutional networks. In: Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems, pp. 397–400 (2018)
16. Chen, K., et al.: Dynamic spatio-temporal graph-based cnns for traffic prediction (2018)
17. Deng, S., Jia, S., Chen, J.: Exploring spatial–temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data. Appl. Soft Comput. **78**, 712–721 (2019)
18. Bargar, A., Gupta, A., Gupta, S., Ma, D.: Interactive visual analytics for multi-city bikeshare data analysis. In: The 3rd International Workshop on Urban Computing (UrbComp 2014), New York, USA, vol. 45 (2014)
19. Dell'Amico, M., Iori, M., Novellani, S., Subramanian, A.: The bike sharing rebalancing problem with stochastic demands. Transp. Res. Part B Methodol. **118**(DEC), 362–380 (2018)
20. Vogel, P., Greiser, T., Mattfeld, D., Sciences, B.: Understanding bike-sharing systems using data mining: exploring activity patterns. Proc. Soc. Behav. Sci. **20**(6), 514–523 (2011)
21. Yan, Y., Tao, Y., Jin, X., Ren, S., Lin, H.: Visual analytics of bike-sharing data based on tensor factorization. J. Visual. **21**(3), 495–509 (2018). https://doi.org/10.1007/s12650-017-0463-1
22. Lihua, N., Xiaorong, C., Qian, H.: ARIMA model for traffic flow prediction based on wavelet analysis. In: The 2nd International Conference on Information Science and Engineering (2011)
23. Zhang, N., Zhang, Y., Lu, H.: Seasonal autoregressive integrated moving average and support vector machine models: prediction of short-term traffic flow on freeways. Transp. Res. Record. **2215**(1), 85–92 (2011)
24. Ahn, J.Y., Ko, E., Kim, E.Y.: Predicting spatiotemporal traffic flow based on support vector regression and Bayesian classifier. In: IEEE Fifth International Conference on Big Data & Cloud Computing (2015)
25. Xie, P., Li, T., Liu, J., Du, S., Zhang, J.: Urban flow prediction from spatiotemporal data using machine learning: a survey. Inform. Fusion. **59**, 1–2 (2020)
26. Cho, K., et al.: Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation (2014)
27. Cho, K., Merrienboer, B.V., Bahdanau, D., Bengio, Y.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches (2014)
28. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)

29. Bahdanau, D., Cho, K., Bengio, Y.: Neural Machine Translation by Jointly Learning to Align and Translate (2014)
30. Fu, R., Zhang, Z., Li, L.: Using LSTM and GRU neural network methods for traffic flow prediction. In: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), pp. 324–328. IEEE (2016)
31. Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., Schmidhuber, N.: LSTM: a search space Odyssey. IEEE Trans. Neural Netw. Learn. Syst. **28**(10), 2222–2232 (2016)
32. Chen, T., Guestrin, C.: XGBoost: a scalable tree boosting system. In: the 22nd ACM SIGKDD International Conference (2016)