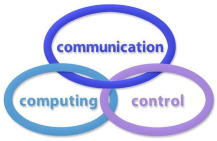
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 15, Issue: 5, Month: October, Year: 2020
Article Number: 3906, https://doi.org/10.15837/ijccc.2020.5.3906




CCC Publications


# A Convolutional Long Short-Term Memory Neural Network Based Prediction Model

Y. H. Tian, Q. Wu, Y. Zhang

**Yonghong Tian***
College of Data Science and Application
Inner Mongolia University of Technology
Hohhot 010080, China
*Corresponding author: tyh@imut.edu.cn

**Qi Wu**
College of Data Science and Application
Inner Mongolia University of Technology
Hohhot 010080, China
wq0103wq@163.com

**Yue Zhang**
College of Data Science and Application
Inner Mongolia University of Technology
Hohhot 010080, China
yueer_zhang@163.com

**Abstract**

In recent years, the market demand for online car-hailing service has expanded dramatically. To satisfy the daily travel needs, it is important to predict the supply and demand of online car-hailing in an accurate manner, and make active scheduling based on the predicted gap between supply and demand. This paper puts forward a novel supply and demand prediction model for online car-hailing, which combines the merits of convolutional neural network (CNN) and long short-term memory (LSTM). The proposed model was named convolutional LSTM (C-LSTM). Next, the original data on online car-hailing were processed, and the key features that affect the supply and demand prediction were extracted. After that, the C-LSTM was optimized by the AdaBound algorithm during the training process. Finally, the superiority of the C-LSTM in predicting online car-hailing supply and demand was proved through contrastive experiments.

**Keywords:** online car-hailing, supply and demand prediction, long short-term memory (LSTM), convolutional neural network (CNN), AdaBound.

# 1 Introduction

With the development of Internet technology, online car-hailing platforms are growing in number and scale. Based on supply-demand information, the platforms connect drivers and passengers with a sharing mechanism called Internet taxi booking. Online car-hailing has become a main travel mode in many countries. Take China for example; there are 392 million online car-hailing users by the end of 2019. Many online car-hailing platforms, namely, DiDi, UCAR and Uber [16], are very popular in urban areas, facilitating daily trips and relieving the traffic pressure.

The boom of online car-hailing platforms also faces a major problem: the supply-demand balance is gradually disrupted in some areas, because users may change the travel mode. Thus, it is urgent to dispatch the drivers in advance to meet the travel demand of passengers. This calls for dynamic forecast of supply and demand. Traditionally, the supply and demand of online car-hailing is predicted by the logit model [1], autoregressive–moving-average (ARMA) model [12] and backpropagation neural network (BPNN) [6]. Yang et al. [19] established a taxi demand model by analyzing the demands of passengers on taxis at different times. Jiang et al. [7] established a model based on the online car-hailing order data on the mobile APP and completed the prediction of online car-hailing. However, the lack of spatiotemporal characteristics may affect the prediction effect of online car-hailing.

Considering data sparsity, Zhang et al. [20] proposed a double ensemble gradient boosting decision tree to predict the supply and demand gap of online car-hailing. Based on nonlinear support vector machine (SVM) and BPNN [3], Lu et al. [10] created a two-stage supply and demand prediction model for online car-hailing, and extracted the features that affect the prediction; however, the BPNN, as a traditional neural network (NN), perform poorly in solving time series problems and converges slowly in the face of largescale data. With the aid of deep learning (DL) technique [13], Xu et al. [18] predicted online car-hailing supply and demand based on long short term memory (LSTM) with sliding window, but wasted too much time in LSTM training.

This paper attempts to solve the defects in the above studies on supply and demand forecast of online car-hailing, and accurately predict the range of supply-demand gap. For these purposes, the convolutional neural network (CNN) [4, 8, 15, 21] and the LSTM were combined into the convolutional LSTM (C-LSTM) model, which was then trained with AdaBound. Firstly, the CNN was adopted to extract the features of supply and demand forecast for online car-hailing, while reducing data size and enhancing model generalization. Since the forecast is a time series problem, the LSTM was employed to arrange the extracted features in chronological order, to complete the forecast of the range of online car-hailing supply-demand. The prediction effect of the proposed C-LSTM model was proved through contrastive experiments.

# 2 The CNN

The CNN is a feedforward NN widely adopted for computer vision [14] and natural language processing [9]. The network structure is translation invariant, and generally composed of one or more convolutional layers, pooling layers and fully-connected layers. In each layer, the neurons are connected locally to extract and transform the hierarchical features of the input; meanwhile, these neurons are also connected to those on the superior layer, with the same connection weight.

In each convolutional layer, several feature maps are generated through convolution by different kernels. Each feature map consists of several neurons arranged in matrix. Every neuron is only connected to some of its neighboring neurons. The neurons in the same feature map share a kernel, which is equivalent to weight. Therefore, the weight sharing strategy can reduce the connections between different layers and lower the risk of overfitting.

The eigenvalues of the supply and demand prediction of online car hailing can be expressed as:

$$G_n = \left[G^1, G^2, \cdots, G^{T_n}\right], n = 1, 2, \cdots, N \tag{1}$$

where, $G^{\mathrm{t}} = \left[G_1^t, G_2^t, \cdots, G_m^t\right]^T \in R^{\mathrm{m}\times 1}$ is the eigenvalue at time $t$; $N$ is the number of eigenvalues. $L$ is the length of the time series of the convolution kernel. Then, the eigenvalue in the period [$t$,

$t+T-1]$ can be described as:

$$G_n^{t:t+T-1} = \left[G^t, G^{t+1}, \cdots, G^{t+T-1}\right]^T \tag{2}$$

The output $\beta_{t+T-1}$ of convolution at time $t+T-1$ can be defined as:

$$\beta_{t+T-1} = \sigma(W_i^T \bullet G_n^{t:t+T-1} + b) \tag{3}$$

where, $\sigma$ is the activation function; $W_i$ is the 1D kernel; $\bullet$ is matrix multiplication; $b$ is the bias term.

In addition to convolution, pooling is another important operation in the CNN. This operation reduces the dimensionality of each feature map, making the computation less complex. The pooling layer, a.k.a. the subsampling layer, either implements mean pooling or max pooling. If max pooling is adopted, the maximum eigenvalue can be obtained as $\hat{G} = \max\{\beta_{t+T-1}\}$.

## 3 The LSTM

The LSTM is a time-recurrent neural network (RNN) proposed by Hochreater et al. [5] to solve the problem of long-term dependency [17]. This NN has been widely applied in machine translation [2], subtitle generation and stock forecast, thanks to its excellence in natural language processing and ability to avoid the vanishing gradient problem.

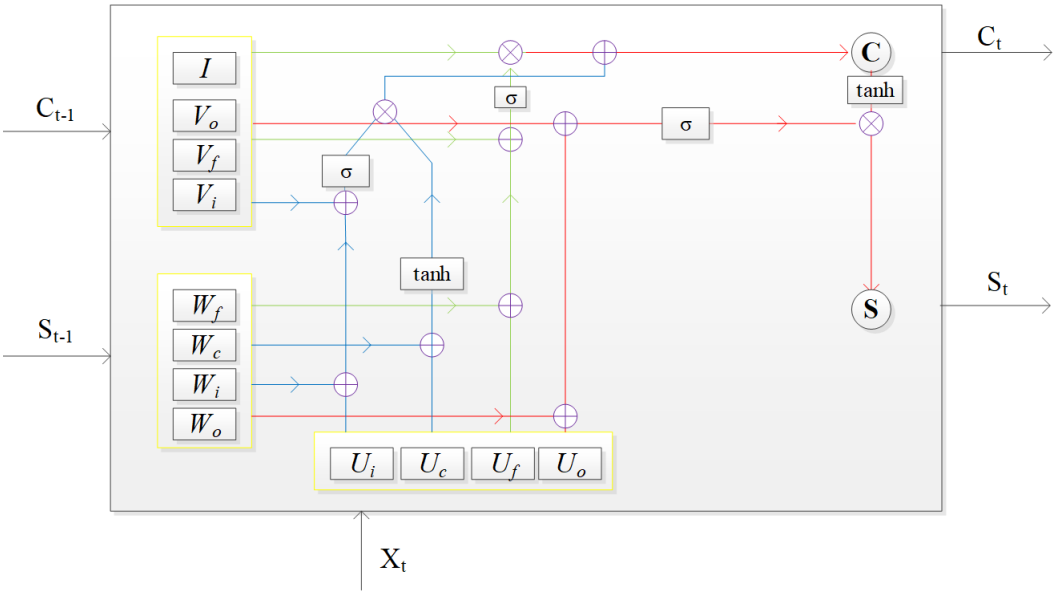



Figure 1: Structure of the LSTM

The LSTM memory unit is mainly composed of four modules: cell status, input gate, output gate and forget gate. The input gate controls the update of memory state; the output gate controls the output in the current state; the forget gate clears useless information. The functions of the three gates are equivalent to write, read and reset operations. Together, the three gates ensure that the state of each moment is stored in the cell. The LSTM structure diagram is shown in Figure 1. The input at time $t$ includes $x_t$, $c_{t-1}$ and $s_{t-1}$, and the output includes $c_t$ and $s_t$, $W_i$, $W_f$, $W_c$ and $W_o$ be the weights of input gate, forget gate, cell, and output gate, respectively, $b_f$, $b_i$, $b_c$, and $b_o$ be the biases of input gate, forget gate, cell, and output gate, respectively. Then, the cell state of the LSTM can be updated as follows:

The main function of the input gate is to determine how much of the input $x_t$ can be retained in the $c_t$. The realization formula is shown in formula (4):

$$\begin{cases} i_t = \sigma\left(U_i \cdot x_t + W_i \bullet s_{t-1} + V_i \cdot c_{t-1}\right) \\ \tilde{c}_t = \tanh\left(U_c \cdot x_t + W_c \cdot s_{t-1}\right) \end{cases} \tag{4}$$

Here $i$ stands for "input"; $i_t$ represents the input of the input gate at time $t$, while $i_t \bigotimes \tilde{c}_t$ represents the components stored in $c_t$ after the input gate, and "$\bigotimes$" represents the multiplication of the corresponding elements in the corresponding vector.

The main function of the forget gate is to determine how many components of $c_{t-1}$ can be retained in $c_t$ at time $t$, and the realization formula is shown in formula (5):

$$f_t = \sigma\left(U_f \cdot x_t + W_f \cdot s_{t-1} + V_f \cdot c_{t-1}\right) \tag{5}$$

The $f$ here stands for "forget"; As with the input gate $\tilde{c}_t$, that is, after forgetting the gate, the component retained in the $c_t$ is $f_t \bigotimes \tilde{c}_{t-1}$.

The output gate determines the components retained in the hidden layer $s_t$ into the output $o_t$ through the control unit $c_t$. State $C$ after the input gate and the forgotten gate, as shown in formula (6):

$$\mathrm{c}_t = i_t \otimes \tilde{c}_t + f_t \otimes \tilde{c}_{t-1} \tag{6}$$

The former term refers to the components retained in the $c_t$ by the input gate, and the latter term refers to the components retained in the $c_t$ by the forgotten gate. Secondly, in order to determine how many components in $c_t$ are retained in $s_t$, the specific implementation is shown in formula (7):

$$o_t = \sigma\left(U_o \bullet x_t + W_o \bullet s_{t-1} + V_o \bullet c_t\right) \tag{7}$$

where, $o_t$ represents the state of the output layer at time $t$. After passing through the output gate, the components retained on the hidden layer are implemented as shown in formula (8):

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{8}$$

LSTM network has been widely used in many fields because it can avoid the problem of gradient vanishing effectively and can process the input of a longer time sequence at the same time.

# 4 Design and optimization of the C-LSTM

Based on the above analysis, this section combines the CNN and the LSTM into the C-LSTM model, and optimizes the model training with AdaBound.

## 4.1 Model design

Figure 2 provides the structure of the C-LSTM model for supply and demand prediction of online car-hailing.

The C-LSTM firstly processes the input data into a matrix of time series. Next, the CNN extracts features and reduces the dimensionality of the input data, creating a low-dimensional feature map which contains the data features. The feature map is then imported to the LSTM, where the extracted features are arranged by temporal relationship. Finally, the LSTM output is normalized by the softmax function, and outputted as a k-dimensional vector. Each dimension of the vector is the probability that the samples fall in one of the k classes preset for predicted values. Here, the supply-demand gap was divided into six classes: GA (0), GB (1-10), GC (10-20), GD (20-30), Ge (30-50) and GF (>50), which correspond to the six dimensions of the output layer.

The probabilities of the $k$ classes add up to 1. The final prediction result can be expressed as:

$$T\left(B_t\right) = MAX\left(P\left(y = m|b_t\right)\right) \tag{9}$$

where, $b_t$ is a sample; $P\left(y = m|b_t\right)$ is the probability for $b_t$ to fall into class $m$.

Formula (9) shows that the maximum of the probabilities for the six classes is chosen as the prediction range of our model.

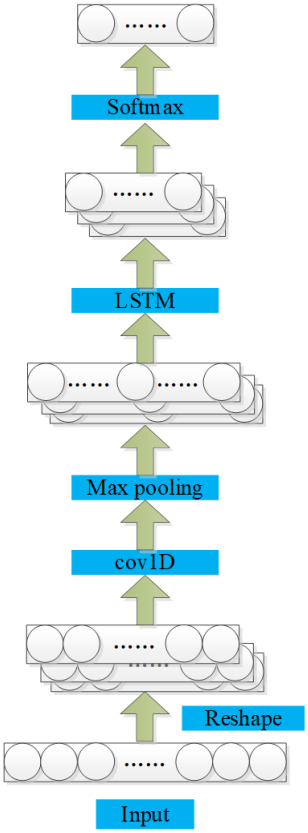



Figure 2: Structure of the C-LSTM

### 4.2 Training optimization

The prediction effect of the C-LSTM model hinges on the training quality. To improve the prediction effect, the training of the C-LSTM was optimized by AdaBound, a desirable tool for computer vision, natural language processing and the DL.

Proposed by Luo et al. [11], AdaBound is a DL optimization algorithm with adaptive learning rate. The algorithm is improved from the Adam optimization algorithm, whose instability and extreme learning rate have led to poor generalization ability.

The AdaBound draws the boundaries of dynamic changes in learning rate, and ensures the gradual and smooth transition from the Adam optimization algorithm to stochastic gradient descent (SDG). Moreover, AdaBound is not very sensitive to hyper-parameters, which saves lots of time for parameter adjustment and further improves the learning effect.

The specific steps of the AdaBound are illustrated below Algorithm 1.

**Algorithm 1** AdaBound

1: Input: $X_1 \in F$, initial step size $\alpha$, $\{\beta_{1t}\}_{t=1}^{T}$, $\beta_2$, lower bound function $\eta_1$, and upper bound function $\eta_u$
2: Set $m_0$=0, $v_0$=0
3: for $t$=1 to $T$ do
4: $g_t = \nabla f_t(X_t)$
5: $m_t = \beta_{1t} m_{t-1} + (1-\beta_{1t})\, g_t$
6: $V_t = \beta_2 V_{t-1} + (1-\beta_2)$ and $V_t = \mathrm{diag}(V_t)$
7: $\hat{\eta}_t = \mathrm{Clip}\left(\alpha/\sqrt{V_t}, \eta_t(t), \eta_u(t)\right)$ and $\eta_t = \hat{\eta}_t/\sqrt{t}$
8: $X_{t+1} = \prod_{F,\mathrm{diag}(\eta_t^{-1})}(X_t - \eta_t \odot m_t)$
9: End for

The AdaBound involves six parameters, namely, the learning rate or step size ($\alpha$), the exponential

decay rate of the first-order moment estimation ($\beta_1$), the exponential decay rate of the second moment estimation ($\beta_2$), the hyper-parameter ($\varepsilon$) that prevents the denominator from being zero, the final learning rate (final_lr), and the convergence rate ($\gamma$). Because the C-LSTM model is mainly trained on Keras, the six parameters above were set to $\alpha$=0.001, ($\beta_1$)=0.9, $\beta_2$=0.999, $\varepsilon$=1e-08, the final-lr=0.1, and $\gamma$=1e-3, respectively.

The AdaBound-based optimization of C-LSTM training is described as follows Algorithm 2.

**Algorithm 2**

1: Input: $m$ training samples; the number of neurons in each hidden layer and the output layer, the total number of layers $h$; the activation function; the loss function; iterative step $s$; maximum number of iterations MAX; the termination threshold $\varepsilon$.
2: Output: weights $W$ and bias $b$ of each hidden layer and the output layer.
3: Initialize weights $W$ and biases $b$ of each hidden layer and the output layer in a random manner.
4: for iter to 1 to MAX:
5: for $i$=1 to $m$: perform forward propagation to calculate the activation value.
6: for $l$=2 to $h$, perform forward propagation to calculate the activation value.
7: Calculate the gradient of the output layer by mean squared loss function.
8: for $l$=$h$ to 2, perform back-propagation to calculate the gradient of each layer.
9: for l=2 to h, update the weights $W$ and biases $b$ of the lst layer using Adam optimization algorithm.
10: If the variations in weights $W$ and biases $b$ are below the termination threshold , go to the next step.
11: Output weights $W$ and biases $b$ of each hidden layer and the output layer.

# 5 Experimental verification

## 5.1 Experimental environment

To verify its performance, the C-LSTM was subjected to experiments on a computer (GPU: GeForce GTX 1060) running on Windows. The software development environment, programming language, and DL framework are PyCharm Community Edition (64bit), Python 3.6 (64bit), and Keras, respectively. Besides, various Python libraries such as sklearn, panda and numpy were installed in the development environment.

## 5.2 Feature selection

Selecting reasonable feature data can effectively improve the training effect of the prediction model. In this section, we will analyze the original data of online car-hailing and select the factors that affect the forecasting research of online car-hailing supply-demand prediction, including weather, temperature, traffic congestion, air Quality, time, area identification and the difference between supply and demand. Next, we will introduce it in detail based on the selected features. When people choose online car-hailing to travel, they will encounter difficulties in taking a taxi. For example, it is difficult to get a car-hailing car when the weather conditions are bad and the temperature is not suitable. It is also very busy during holiday peak hours and commuting hours It is difficult to get a car; and in traffic congestion and good weather, people may not choose to ride a car online; in addition, Passengers who take online ride-hailing in the previous period will also affect the status of passengers who take a taxi at the current time. Therefore, there are many factors that influence people to travel by online car-hailing.

## 5.3 Experimental data

The experimental data on online car-hailing were provided by DiDi. The data from February 23 to March 17 were selected as the experimental dataset. Each day was divided into 144 10min-long time slices. In this way, a total of 200,448 pieces of data was obtained for our experiments. The experimental dataset was split into a training set and a test set at the ratio of 4:1.

For convenience, the raw data were converted to the CSV format and stored in the MongoDB, a document database. Then, the database was used to view, manage and query the data on online car-hailing. The processed data were stored in CSV format.

The experimental dataset covers multiple features, including regional feature, climate feature, traffic congestion feature, period feature, and supply-demand gap feature.

(1) Regional feature

The regional feature reflects the influence of different regions on supply and demand prediction. The data on this feature were provided by DiDi. The target city was divided into 58 areas of equal size. Hence, the hash tags of the 58 areas were converted in turn into numerical tags and stored in CSV format.

(2) Climate feature

There are three kinds of climate feature: weather feature, temperature feature and PM2.5 feature. These features were extracted with the aid of the MongDB. The temperature, weather, and PM2.5 of each area were queried, recorded every 10min on each day, and stored in CSV format.

(3) Traffic congestion feature

This feature reflects the number of road segments with each degree of congestion. The traffic data in MongoDB were used to write a program to convert the scattered data (Table 1) into integrated congestion information. The processed data were recorded every 10min on each day, and stored in CSV format. The scattered data can be converted by:

$$total\_level = \sum_{\mathrm{n}=1}^{4} C\_leveln * \mathrm{n} \tag{10}$$

(4) Period feature

The online car-hailing situation varies from period to period. Our experimental data were collected on 18 workdays and 6 weekends. In general, more people travel in morning peak hours on weekdays, and the entertainment and dining hours on weekends. Therefore, each workday was split into three periods: morning and evening peak hours (7: 00 9: 00 and 17: 00 19: 00), working hours (9: 00 17: 00), and the remaining hours; each weekend day was split into two periods: peak hours (10: 00 14: 00 and 17: 00 21: 00) and the remaining hours. The features of the simultaneous periods were differentiated by numerical values. The data were recorded every 10min on each day, and stored in CSV format.

(5) Supply-demand gap feature

The supply-demand gap of online car-hailing is affected by the gap value at the previous moment. Thus, the gap value at the previous moment was taken as the feature of the supply-demand gap. The gap value at time $t-1$ is denoted as Gt-1. Considering waiting time and other factors, the gap values at time t-3 to time $t-1$ were selected. The data were recorded every 10min on each day, and stored in CSV format.

After the features were constructed, the various feature data were fused and stored in CSV format. The final feature dataset is described in Table 1, where ID is the regional feature, weather is the weather feature, temp is the temperature feature, PM is the PM2.5 feature, traffic is the traffic congestion feature, period is the period feature, and Gt-3, Gt-2 and Gt-1 are the supply-demand gap feature; the time is the duration of feature collection (10min).

### 5.4 Evaluation indices

The supply and demand prediction of online car-hailing is a multi-classification problem. Therefore, the predictive ability of our C-LSTM model was evaluated by Precision, Recall and F1:

$$Precison = \frac{TP}{TP + FP} \tag{11}$$

$$Recall = \frac{TP}{TP + FN} \tag{12}$$

$$F1 = \frac{2 \times Precison \times Recall}{Precison + Recall} \tag{13}$$

Table 1: Feature dataset

| Time | ID | Weather | Temp | PM | Traffic | Period | $G_{t-3}$ | $G_{t-2}$ | $G_{t-1}$ |
|---|---|---|---|---|---|---|---|---|---|
| 13:00:00-13:10:00 | 21 | 1 | 13 | 80 | 6634 | 0.3 | 9 | 10 | 7 |
| 13:10:00-13:20:00 | 21 | 1 | 13 | 80 | 6252 | 0.3 | 10 | 7 | 16 |
| 13:20:00-13:30:00 | 21 | 1 | 13 | 80 | 6370 | 0.3 | 7 | 16 | 19 |
| 13:30:00-13:40:00 | 21 | 1 | 14 | 80 | 6005 | 0.3 | 16 | 19 | 25 |
| 13:40:00-13:50:00 | 21 | 1 | 14 | 69 | 6372 | 0.3 | 19 | 25 | 50 |
| 13:50:00-14:00:00 | 21 | 1 | 14 | 69 | 6375 | 0.3 | 25 | 50 | 32 |
| 14:00:00-14:10:00 | 21 | 1 | 14 | 69 | 6099 | 0.3 | 50 | 32 | 31 |
| 14:10:00-14:20:00 | 21 | 1 | 14 | 69 | 6516 | 0.3 | 32 | 31 | 24 |
| 14:20:00-14:30:00 | 21 | 1 | 14 | 69 | 6131 | 0.3 | 31 | 24 | 19 |
| 14:30:00-14:40:00 | 21 | 1 | 14 | 69 | 6475 | 0.3 | 24 | 19 | 12 |
| 14:40:00-14:50:00 | 21 | 1 | 14 | 68 | 6446 | 0.3 | 19 | 12 | 25 |
| 14:50:00-15:00:00 | 21 | 1 | 14 | 68 | 6148 | 0.3 | 12 | 25 | 20 |
| 15:00:00-15:10:00 | 21 | 1 | 14 | 68 | 6389 | 0.3 | 25 | 20 | 31 |
| 15:10:00-15:20:00 | 21 | 1 | 14 | 68 | 6262 | 0.3 | 20 | 19 | 31 |
| 15:20:00-15:30:00 | 21 | 1 | 14 | 68 | 6126 | 0.3 | 19 | 31 | 101 |
| 15:30:00-15:40:00 | 21 | 1 | 14 | 68 | 6334 | 0.3 | 31 | 101 | 76 |
| 15:40:00-15:50:00 | 21 | 1 | 14 | 68 | 6985 | 0.3 | 101 | 76 | 10 |
| 15:50:00-16:00:00 | 21 | 1 | 14 | 68 | 6817 | 0.3 | 76 | 10 | 16 |

where, $TP$ is true positive (a result indicating that a given condition is present when it is); $FP$ is false positive (a result indicating that a given condition is present when it is not); $FN$ is false negative (a result indicating that a given condition is not present when it is). Besides, Accuracy was selected to measure the prediction accuracy of our model:

$$Accuracy = \frac{Numberofcorrectpredictions}{Numberoftotalpredictions} \tag{14}$$

### 5.5 Parameter settings

The most important part of the structure design of network model is the design of hidden layer. The number of layers and the parameters of each layer should be determined. The C-LSTM neural network in this paper includes four parts: input layer, convolutional network part, LSTM, and output layer. For the convolutional network part of the model, select Max pooling method and activate function select Relu. The size and number of layers of CNN convolution kernel need to be optimized and adjusted. In the following, the size and number of layers of convolution kernel will be introduced through experiments, and Accuracy will be used as the evaluation standard.

For the selection of the size of the convolution kernel, we simulate the method of extracting features that is common in time series processing, that is, extracting features of different time series by setting the convolution kernel of different sizes. The experimental results of setting the sizes and layers of different convolution kernels are shown in table 2. As can be seen from Table 2, when the number of layers is 2 and the size of the convolution kernel is 64,32, the prediction accuracy of the model is the highest. The experimental results of the number of LSTM nodes are shown in Table 3. It can be seen from the table that when the number of nodes is 32, the prediction accuracy of the model is the highest.

The C-LSTM parameters were adjusted through repeated tests. The final parameter settings are listed in Table 4. It can be seen that the CNN contains two convolutional layers. The first convolution layer has 64 neurons, and the second layer has 32 neurons. The activation functions of the two layers are both rectified linear unit (ReLU). Each convolutional layer is followed by a pooling layer. There are two LSTM layers in our model, each with 32 neurons. The last layer is a dense layer with softmax as the activation function. This layer outputs the predicted result.

Table 2: Experimental results of different convolution kernel size and convolution layer number

| CNN layer | Convolution kernel size | Accuracy |
|---|---|---|
| 2 | (32,1), (16,1) | 0.7219 |
| | (64,1), (16,1) | 0.7217 |
| | (64,1), (32,1) | 0.7246 |
| | (128,1), (64,1) | 0.7220 |
| 3 | (128,1), (64,1), (32,1) | 0.7155 |

Table 3: Experimental results of different number of LSTM nodes

| LSTM nodes | Accuracy |
|---|---|
| 16 | 0.7237 |
| 32 | 0.7246 |
| 64 | 0.7212 |
| 128 | 0.7221 |

Table 4: Parameter settings

| Layers | Type | Output | Activation functions |
|---|---|---|---|
| 1 | Conv1D | (None, 9, 64) | ReLU |
| 2 | MaxPooling1D | (None, 4, 64) | |
| 3 | Conv1D | (None, 2, 32) | ReLU |
| 4 | MaxPooling1D | (None, 1, 32) | |
| 5 | LSTM | (None, 1, 64) | tanh |
| 6 | LSTM | (None, 1, 64) | tanh |
| 7 | Dense | (None, 6) | softmax |

### 5.6 Results analysis

To prove its superiority in predicting supply and demand of online car-hailing, our C-LSTM model was compared with other prediction algorithms on actual data. Besides, the improving effect of AdaBound on C-LSTM prediction was contrasted with that of other optimization algorithms.

(1) Prediction effect of the C-LSTM

The prediction effect of the C-LSTM was compared with that of the classic LSTM, random forest (RF) and the SVM. The effect of each model was measured by Precision, Recall, F1 and Accuracy. To eliminate stochasticity, 30 comparative experiments were carried out, and the mean values of these experiments were taken as the final results (Table 5).

Table 5: Prediction results of different models

| Algorithms \ Indices | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| C-LSTM | 0.7253 | 0.719 | 0.725 | 0.714 |
| LSTM | 0.7139 | 0.706 | 0.714 | 0.703 |
| RF | 0.6893 | 0.687 | 0.691 | 0.683 |
| SVM | 0.6784 | 0.627 | 0.679 | 0.637 |

As shown in Table 5, the proposed C-LSTM achieved greater Precision, Recall, F1 and Accuracy than the three contrastive prediction models, an evidence of its excellence in the forecast of supply and demand of online car-hailing. Besides, the DL models (LSTM and C-LSTM) both outperformed the RF and SVM, indicating that DL is more suitable for our problem than machine learning. To sum up, our model has a good prediction effect of online car-hailing supply and demand.

(2) Analysis of Loss value convergence under different optimizers

In this section, the experiment mainly verifies that the AdaBound algorithm improves the convergence rate of the predictive model training. The optimization effect of AdaBound algorithm will be analyzed through the convergence of functions in the process of model training, and the AdaBound algorithm will be compared with the traditional SGD algorithm and RMSprop algorithm.

The experimental results show that the model cost function of AdaBound algorithm converges most quickly in the process of model training, while the traditional SGD algorithm and RMSprop algorithm converge more slowly. At the same time, the cost function value of AdaBound algorithm is lower than SGD algorithm and RMSprop algorithm. In the training process of the model, the smaller the loss value is, the better the learning effect of the model will be (See Figure 3).

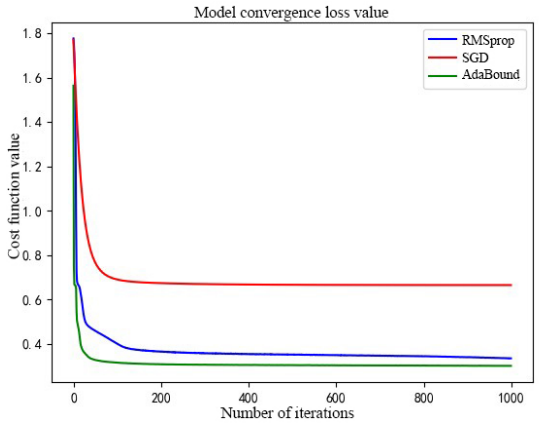



Figure 3: Loss convergence comparison of different optimizer

(3) Optimization effect of AdaBound

This subsection aims to verify the advantages of AdaBound in optimizing the training of the C-LSTM model. For comparison, the Adam optimization algorithm and root mean square prop (RMSprop) were separately introduced to train the C-LSTM. The following parameters were set to the same values as those of AdaBound: step size ($\alpha$), the exponential decay rate of the first-order moment estimation ($\beta_1$), the exponential decay rate of the second moment estimation ($\beta_2$), the hyper-parameter ($\varepsilon$) that prevents the denominator from being zero. Besides, 30 comparative experiments were carried out, and the mean values of these experiments were taken as the final results (Table 6).

Table 6: Prediction results of the C-LSTM trained by different algorithms

| Algorithms \ Indices | Accuracy | Precision | Recall | F1 |
|---|---|---|---|---|
| AdaBound | 0.7253 | 0.719 | 0.725 | 0.714 |
| Adam | 0.7134 | 0.705 | 0.713 | 0.708 |
| RMSprop | 0.7006 | 0.698 | 0.701 | 0.701 |

Table 6 shows that the C-LSTM trained by AdaBound had better indices than that trained by other optimization algorithms. This means the AdaBound can greatly improve the prediction effect of the C-LSTM model on online car-hailing supply and demand.

## 6 Conclusions

This paper puts forward a novel model called the C-LSTM to predict the supply and demand of online car-hailing. Firstly, the CNN and LSTM were subjected to structural analysis, and combined into the C-LSTM model. Next, the online car-hailing data provided by DiDi were processed, and divided into a training set and a test set. After that, the C-LSTM model was optimized by the AdaBound algorithm to improve its prediction effect. The superiority of the C-LSTM in supply and

demand prediction of online car-hailing was confirmed through contrastive experiments. The research results provide a reference for online scheduling of cars, and help to reduce the waiting time of users, making online car-hailing platforms more efficient. The future research will further optimize the structure of the C-LSTM model based on the extended versions of the LSTM.

### Funding

The work was supported by the Natural Science Foundation of Inner Mongolia (Grant No.: 2013MS0920), and Science and Technology Planning Project of Inner Mongolia (Grant No.: 201502015).

### Conflict of interest

The authors declare no conflict of interest.

## References

[1] Castro, M.; Martínez, F.; Munizaga, M.A. (2013). Estimation of a constrained multinomial logitmodel, *Transportation*, 40(3), 563–581, 2013.

[2] Cui, Y.M.; Wang, S.J; Li, J.F. (2016). *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Association for Computational Linguistics, San Diego, California, 2016.

[3] Gorur, K.; Bozkurt, M.R.; Bascil, M.S.; Temurtas, F. (2019). GKP signal processing using deep CNN and SVM for tongue-machine interface, *Traitement du Signal*, 36(4), 319–329, 2019.

[4] Goyat, R.; Kumar, G.; Rai, M.K.; Saha, R. (2019). Implications of blockchain technology in supply chain management, *Journal of System and Management Sciences*, 9(3), 92–103, 2019.

[5] Hochreiter, S.; Schmidhuber, J. (1997). Long short-term memory, *Neural Computation*, 9(8), 1735–1780, 1997.

[6] Huang, Q.; Cui, L.M. (2019). Design and application of face recognition algorithm based on improved backpropagation neural network, *Revue d'Intelligence Artificielle*, 33(1), 25–32, 2019.

[7] Jiang, W.; Wo, T.; Zhang, M.; Yang, R. (2015). A method for private car transportation dispatching based on a passenger demand model, *International Conference on Internet of Vehicles*, 37–48, 2015.

[8] Kim, B.S.; Kim, T.G. (2019). Cooperation of simulation and data model for performance analysis of complex systems, *International Journal of Simulation Modelling*, 18(4), 608–619, 2019.

[9] Li, C.; Zhan, G.; Li, Z. (2018). News text classification based on improved Bi-LSTM-CNN, *2018 IEEE 9th International Conference on Information Technology in Medicine and Education (ITME)*, 890–893, 2018.

[10] Lu, L.; Lai, X.F.; Li, F. (2019). Forecasting the gap between demand and supply of e-hailing vehicle in large scale of network based on two-stage model, *IEEE Intelligent Transportation Systems Conference*, Auckland, New Zealand, New Zealand, 3880–3885, 2019.

[11] Luo, L.; Xiong, Y.; Liu, Y.; Sun, X. (2019). Adaptive gradient methods with dynamic bound of learning rate, *International Conference on Learning Representations (ICLR), 2019*.

[12] Moreira–Matias, L.; Gama, J.; Ferreira, M.; Mendes-Moreira, J.; Damas, L. (2013). Predicting taxi passenger demand using streaming data, *IEEE Transactions on Intelligent Transportation Systems*, 14(3), 1393–1402, 2013.

[13] Neelapu, R.; Devi, G.L.; Rao, K.S. (2018). Deep learning based conventional neural network architecture for medical image classification, *Traitement du Signal*, 35(2), 169–182, 2018.

[14] Nishani, E.; Çiço, B. (2017). Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation, *2017 IEEE 6th Mediterranean Conference on Embedded Computing (MECO)*, Bar, Montenegro, 1–8, 2017.

[15] Wajeed, M.A.; Sreenivasulu, V. (2019). Image based tumor cells identification using convolutional neural network and auto encoders, *Traitement du Signal*, 36(5), 445–453, 2019.

[16] Wang, M.S.; Mu, L. (2018). Spatial disparities of Uber accessibility: An exploratory analysis in Atlanta, USA, *Computers, Environment and Urban Systems*, 67, 169–175, 2018.

[17] Williams, R.J.; Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks, *Neural Computation*, 1(2), 270–280, 1989.

[18] Xu, J.; Rahmatizadeh, R.; Boloni, L. (2017). A sequence learning model with recurrent neural networks for taxi demand prediction, *IEEE Conference on Local Computer Networks*, Singapore, 261–268, 2017.

[19] Yang, C.; Gonzales, E.J. (2014). Modeling taxi trip demand by time of day in New York city, *Transportation Research Record*, 2429(1), 110–120, 2014.

[20] Zhang, X.; Wang, X.; Chen, W. (2017). A Taxi gap prediction method via double ensemble gradient boosting decision tree, *IEEE International Conference on Big Data Security on Cloud*, Beijing, China, 26–28, 2017.

[21] Zhang, Z.; Guan, Z.L.; Zhang, J.; Xie, X. (2019). A novel job-shop scheduling strategy based on particle swarm optimization and neural network, *International Journal of Simulation Modelling*, 18(4), 699–707, 2019.


Journal's webpage: http://univagora.ro/jour/index.php/ijccc/


This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

*Cite this paper as:*

Tian, Y. H.; Wu, Q.; Zhang, Y. (2020). A Convolutional Long Short-Term Memory Neural Network Based Prediction Model, *International Journal of Computers Communications & Control*, 15(5), 3906, 2020.

https://doi.org/10.15837/ijccc.2020.5.3906