

Research Article

Traffic Status Prediction of Arterial Roads Based on the Deep Recurrent Q-Learning

Wei Hao ¹, Donglei Rong¹, Kefu Yi ², Qiang Zeng ³, Zhibo Gao ⁴, Wenguang Wu ²,
Chongfeng Wei⁵ and Biljana Sceanovic⁶

¹Hunan Key Laboratory of Smart Roadway and Cooperative Vehicle-Infrastructure Systems, Changsha University of Science and Technology, Changsha, Hunan 410205, China

²School of Automotive and Mechanical Engineering, Changsha University of Science and Technology, Changsha 410114, China

³School of Civil Engineering and Transportation, South China University of Technology, Guangzhou 510641, China

⁴Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, 4800 Cao'an Road, Shanghai, China

⁵Mechanical and Construction Engineering, Northumbria University, Ellison Place, Newcastle upon Tyne NE1 8ST, UK

⁶Faculty of Civil Engineering, University of Montenegro, 81000 Podgorica, Montenegro

Correspondence should be addressed to Kefu Yi; corfyi@csust.edu.cn, Qiang Zeng; zengqiang@scut.edu.cn, Zhibo Gao; gaozhibo@tongji.edu.cn, and Wenguang Wu; wwgllq@csust.edu.cn

Received 27 May 2020; Revised 13 July 2020; Accepted 7 September 2020; Published 19 September 2020

Academic Editor: Yanyong Guo

Copyright © 2020 Wei Hao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

With the exponential growth of traffic data and the complexity of traffic conditions, in order to effectively store and analyse data to feed back valid information, this paper proposed an urban road traffic status prediction model based on the optimized deep recurrent Q-Learning method. The model is based on the optimized Long Short-Term Memory (LSTM) algorithm to handle the explosive growth of Q-table data, which not only avoids the gradient explosion and disappearance but also has the efficient storage and analysis. The continuous training and memory storage of the training sets are used to improve the system sensitivity, and then, the test sets are predicted based on the accumulated experience pool to obtain high-precision prediction results. The traffic flow data from Wanjiali Road to Shuangtang Road in Changsha City are tested as a case. The research results show that the prediction of the traffic delay index is within a reasonable interval, and it is significantly better than traditional prediction methods such as the LSTM, K-Nearest Neighbor (KNN), Support Vector Machines (SVM), exponential smoothing method, and Back Propagation (BP) neural network, which shows that the model proposed in this paper has the feasibility of application.

1. Introduction

With the development of urbanization, there is a prominent contradiction between the transportation infrastructure and the vehicle population, and the problem of traffic congestion has become more serious, which inevitably leads to the increasing of travel time, intensified environmental pollution, and economic loss [1]. Prevention is the first way to control traffic congestion. According to the existing traffic states, the changing trend in a short time is predicted, and then, the information platform is used to issue an early warning to divert the traffic to avoid or ease congestion

[2–4]. Therefore, how to establish a long-term model for timely warning of traffic congestion is the research focus of urban intelligent transportation system optimization [5–7].

A variety of methods, including the time series, machine learning, and artificial neural networks, have been proposed for traffic congestion prediction. Since the time-series characteristics of traffic flow data were discovered [8], some scholars used autoregressive differential moving average models [9] to predict the traffic flow on expressway [10, 11]. Because the temporal distribution of traffic flow data is interrelated, some scholars used nonparametric regression methods to build macrotraffic models and found that the

prediction result is better than time-series algorithms [12–14]. However, these methods based on statistics and traffic models require a large amount of historical data and construct many assumptions, so they are difficult to apply to nonlinear traffic flow [15–18].

In recent years, machine learning algorithms, such as the back propagation neural network [19, 20], have gradually been used in traffic prediction with the advantage to handle the nonlinearity problems. Because of the long training time of the back propagation neural network and the tendency to fall into the local optimum, some scholars also used the Support Vector Machine (SVM) [21–23] and K-Nearest Neighbor (KNN) [24–26] to predict the traffic status. Moreover, some scholars found that the time series of short-term traffic flow has chaotic characteristics. To deal with the abovementioned issues, many methods, such as combined vector machine-based [27] and phase space reconstruction-based [28], have been proposed to achieve better results. However, most of these machine learning-based methods lack robustness to catch the huge data, resulting in the model generally lacking long-term effectiveness and scalability [29–31].

Facing on the lots of traffic flow data, scholars have gradually turned to use the deep learning method, a learning algorithm that can simulate the multilayered perceptual structure of the human brain to recognize the data patterns. At present, breakthroughs have been made in many fields such as computer vision, speech recognition, and natural language processing. Deep learning has gradually been adopted by Stanford University, Google, Baidu Research Institute, and other authoritative organizations with the strategic direction for the development of data mining and artificial intelligence [32, 33]. Kuremoto et al. [34] combined the restricted Boltzmann machine with the time-series laws to obtain a prediction model, which fits the sample data with the minimum model energy. Lv et al. [35] proposed a deep learning model to predict traffic flow based on an automatic coding network using compression coding in the input data. Zhao et al. [36] proposed a traffic congestion prediction model based on the improved SVM, which can learn the characteristics of traffic flow parameters through the deep structure by digitizing different environmental and human factors. The abovementioned methods speed up data processing by applying the deep learning models but do not take into account the dimensional disaster caused by the high-dimensional states of traffic flow parameters. To address the abovementioned problems, some scholars used data compression technology based on the LSTM, Principal Component Analysis (PCA) [37], CUR matrix decomposition algorithm [38], and Discrete Cosine Transform (DCT) method [39] to perform data dimension reduction.

Q-Learning can efficiently store and extract data to provide support for traffic prediction. The LSTM network reduces the frequency of gradient explosion and disappearance, so it is suitable for capturing the spatiotemporal evolution of traffic state parameters [40–43]. In this paper, considering the time sequence of traffic flow parameters and the continuity of traffic congestion effects, the recurrent neural network model is used to train the extracted features

and to obtain low-dimensional vectors of historical information, and then, the resulting vectors are stitched to achieve classification training. Finally, an urban road traffic state prediction model based on the optimized deep recurrent Q-Learning method is established. The model proposed in this paper has the following contributions:

- (1) The model effectively solves the problem of gradient explosion and gradient disappearance in the prediction process of LSTM
- (2) The model effectively extracts the associated features of the traffic data, so it has better prediction efficiency and accuracy
- (3) The model will provide a feasible prediction method for the construction of an intelligent transportation system due to its efficiency and feasibility

The rest of this paper is organized as follows. Section 2 points out the problems to be solved and the corresponding methods in this paper. Then, Sections 3 and 4 lead to the principles and steps of the Q-Learning and the LSTM. After that, the deep recurrent Q-Learning network model is constructed in Section 5. Besides, the example analysis in Section 6 proves the stability and feasibility of the method. Finally, Section 7 concludes the paper.

2. Specific Problems and Solutions

2.1. Specific Problems. The problems with urban traffic data are high repeatability, high loss rate, and poor correlation. The existing prediction methods mainly discuss the results of independent analysis and whether they meet the needs of further verification. Therefore, the following problems exist in data preprocessing and optimization prediction.

Regarding the problem of data relevance: the relationship between the states at the previous moment and the next moment lacks effective connection. Therefore, the information at different states is disconnected, and the timeliness of the data cannot be fully exerted. As a result, the prediction results are not sufficiently correlated with the data at the previous moment and lack of persuasiveness.

Speaking of the problem of data storage: based on the existing analysis methods, the storage capacity of the database will quickly reach the threshold, which is not conducive to long-term and durable prediction. Besides, repeated analysis steps will increase the feedback delay and cannot fulfill the requirements of low-latency traffic prediction.

Concerning the problem of comprehensive data analysis: the existing analysis focuses on fixed types of data, and the traffic environment is an integrated system. Therefore, even if the prediction results are accurate, they cannot reflect the objective situation.

2.2. Solutions. For the abovementioned three research problems, this paper will propose the corresponding solutions:

For the problem of data relevance: based on the optimized LSTM model, the effective correlation and

information accumulation of different data types are strengthened, and the correlation degree of data at different moments is strengthened.

For the problem of data storage: the Q-Learning functionalizes the data information, and each data cell can be realized by the expression of functions. This method not only reduces the pressure of data storage but also improves analysis efficiency and accuracy.

For the problem of comprehensive data analysis: the traffic conditions are affected by multiple factors. Therefore, when selecting the characteristic data types, in addition to the basic parameters of traffic flow, climate and temperature are also considered. That is to establish a multidimensional data analysis system, making the prediction results more accurate and objective.

3. Q-Learning Principle and Application Steps

The steps of the Q-Learning are listed as follows: the state of the agent in the environment E is S , and the actions taken by the agent constitute the action space A . It takes different actions to transfer between states, and the reward function obtained is R . To achieve the optimal strategy, the Q-Learning estimates the value of each action choice in each state. The Q-Learning uses $Q(S, A)$ to represent the value function of state-to-action and continuously updates the value of $Q(S, A)$ according to the state transition. Finally, the Q-Learning obtains the optimal strategy based on $Q(S, A)$.

The value function $Q(S, A)$ of the traffic state is updated as follows: assuming the state of the agent at time t is s , the action is a_t . Then, the state transitions to time $t + 1$, the state is s_{t+1} , and the reward is r_t . Finally, the agent updates the value of $Q(s, a)$ according to all records (s_t, a_t, r_t, s_{t+1}) to find the optimal strategy. The corresponding update function is shown in the following equation:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \lambda \max_a Q(s_{t+1}, a) - Q(s_t, a_t) \right], \quad (1)$$

where $Q(s_t, a_t)$ is the current Q-table, α is the learning rate, r_{t+1} is the benefit at the next moment, λ is the greed coefficient, and $\max_a Q(s_{t+1}, a)$ is the best benefit in memory.

The deep Q-Learning network combines deep learning and Q-Learning. The network uses the perceptive ability of the deep learning to transform the state to high dimensions and uses the decision-making ability of Q-Learning to map the high-dimensional state representation to the low-dimensional action space [44, 45]. In the Q-Learning algorithm, the table is used to store the value of $Q(s, a)$. In the deep Q-Learning, the state dimension of the agent is high, and the table obviously cannot meet the demand. This problem is solved by using $f(s, a)$ to approximate $Q(s, a)$ [46, 47]. Therefore, based on the corresponding value function neural network model, approximate values can be obtained, thereby reducing the storage pressure of the Q-table and providing ideas and methods for Q-Learning to be applied to traffic state prediction. Finally, the network obtains the action value of congestion and dissipation according to the accumulated experience pool. Figure 1

shows a schematic diagram of the principle of approximating the value of “state-action” through the neural network.

The network helps solve the problems of processing huge data volumes. Due to the strong time series of traffic data, the application of this network will make the analysis results more reliable. Further demonstrations and experiments will be discussed in the following sections.

4. Recurrent Neural Network LSTM Algorithm

4.1. Overview of the Recurrent Neural Network. The recurrent neural network is one of the optimized variants of deep neural networks. It is characterized by the output of the neurons at a certain moment as part of the input of the next moment, and the neural network has the function of memorizing the information of the previous moment which can realize the persistence of the information. As shown in Figure 2, the neural network reads the input x_t of the current time t and obtains the output h_t . At the same time, the information status is returned to the neural network as one of the inputs at the next time point. In order to show the execution action more intuitively, we express it by

$$h_t = f(h_{t-1}, x_t). \quad (2)$$

The output h_t at each moment is related to the input h_{t-1} at the previous moment. The recurrent neural network is the most natural structure for processing sequence data which is exactly what we need to handle historical data and real-time data in this paper.

x_t : the input at time t ,
 $X = [x_1, \dots, x_{t-1}, x_t, x_{t+1}, \dots, x_T]$ is the input sequence.

s_t : the state of the hidden layer at time t , also known as the memory unit of the recurrent neural network.

h_t : the output at time t ,
 $H = [h_1, \dots, h_{t-1}, h_t, h_{t+1}, \dots, h_T]$ is the output sequence.

U : the weight parameter matrix of input sequence information X to hidden layer state S .

W : the weight parameter matrix between the hidden layer states S .

V : the weight parameter matrix of hidden layer state S to output sequence information H .

4.2. Recurrent Neural Network LSTM. If the dependency interval between sequences is long, the gradient disappearance of traffic data will happen in ordinary RNN which is difficult to retain the information at earlier times. The LSTM network remembers long-term historical information through the design of the network structure where the output of the network at time $t + 1$ is applied to itself at time t to avoid the gradient disappearance. Its network expands along the time axis. The schematic diagram and the detailed diagram of the three-layer gate are shown in Figures 3 and 4.

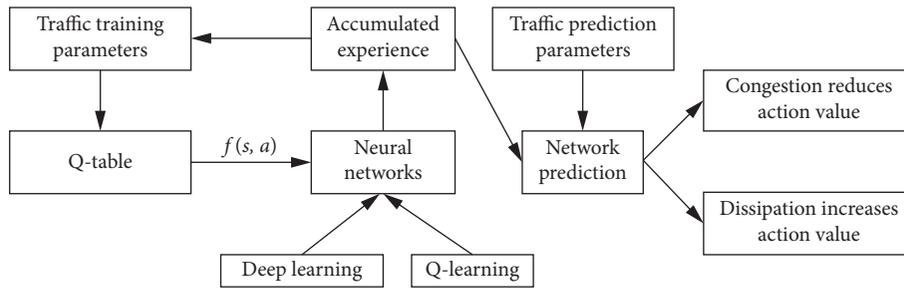


FIGURE 1: Schematic diagram of the neural network approximate establishment of “state-action” value.

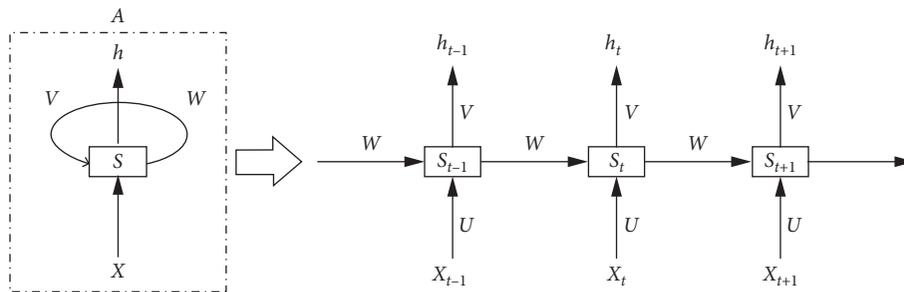


FIGURE 2: Chain structure of the recurrent neural network.

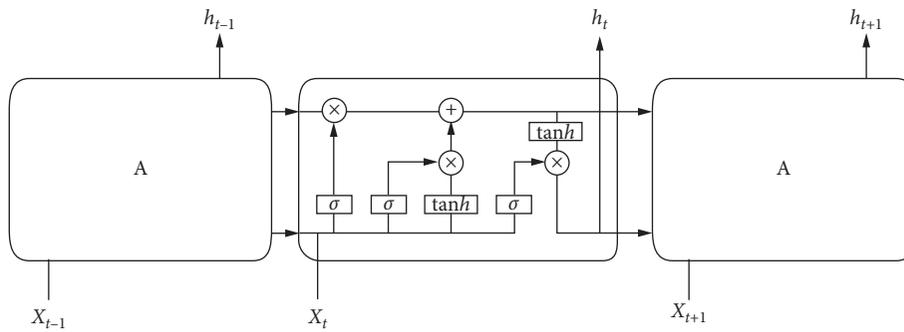


FIGURE 3: The schematic diagram of LSTM expansion by the time axis.

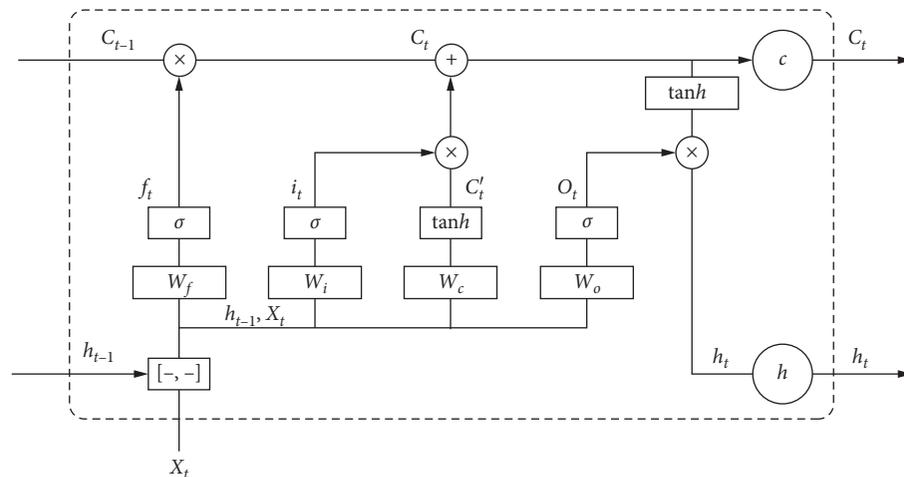


FIGURE 4: Detailed diagram of three-gate levels.

It can be seen from Figure 3 that the LSTM defines the key concept of cell state with the horizontal line. There is less information interaction in the cell with the purpose of memorizing long-term information achieved through cell transmission. For Figure 4, it is made by three-gate layers with the first one is the forget gate. This gate is determined based on the input of the current moment and the output of the previous moment and, then, passes through a *Sigmoid* layer to obtain the results. It determines how much of the cell states from the previous moment is retained to the current moment. The expression function of the forget gate is shown in the following equation:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \quad (3)$$

where f_t represents the output of the forget gate and σ represents the *Sigmoid* function. W_f and b_f represent the weight matrix and the bias term, respectively. $[h_{t-1}, x_t]$ represents the connection of two vectors into a longer vector.

The second one is the input gate, which determines how much of the network input is saved to the cell state at the current moment. The expression functions of the input gate are shown in the following equations:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \quad (4)$$

$$\tilde{c}_t = \tan h(W_c \cdot [h_{t-1}, x_t] + b_c), \quad (5)$$

where c_t is calculated by multiplying the last element state c_{t-1} by the element forget gate f_t , then multiplying the element state c_t by element by the input gate i_t , and finally, adding the two products.

The cell information can be updated based on the results of forget gate and output gate. It is listed as follows:

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t. \quad (6)$$

The last one is the output gate, which controls how many cell states output to the current output value of the LSTM. From Figure 4, the output gate is composed of two parts, one is the state of the cells processed by $\tan h$, and the other is the input information processed by *Sigmoid*. The functions of the output gate are listed in the following equations:

$$\sigma_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \quad (7)$$

$$h_t = o_t \cdot \tan h(c_t), \quad (8)$$

where o_t represents the output of the output gate. W_o and b_o represent the weights and offsets, respectively.

5. Deep Recurrent Q-Learning Network

5.1. State Space. If the amount of acquired data is not large, the Q-Learning can perform data storage and processing efficiently. If the data is large, Q-Learning cannot traverse all states, and there is no such large space to install the Q-value table in memory. Therefore, this paper uses the LSTM model to generalize the states and uses the recurrent neural network to learn the state function. Through continuous deep

reinforcement learning, the model obtains features to describe the current state, while accumulating experience pool. In constructing the state space, it is divided into two steps: state discretization and value evaluation.

Step 1: what the neural network wants to output is the training value under each state, which represents the measure of the pros and cons of developing from this state. The characteristics of the current state S are speed v , delay time d , travel time m , temperature t , and precipitation probability p . If the characteristics of the next state S' are speed v' , delay time d' , travel time m' , temperature t' , and precipitation probability p' , then the corresponding selection behavior of reward accumulation is [speed v , delay time d , travel time m , temperature t , precipitation probability p] minus [speed v' , delay time d' , travel time m' , temperature t' , precipitation probability p']. The resulting values for each position are positive 1 and negative 0 to discretize the behavior (the range is $0 \sim 2^5$, and the selected behavior discretization vector $[0, 1, 0, \dots]$ is transformed into an integer which represents the dimension of the output vector). Based on the results of the abovementioned rewards, the traffic information with better benefits is accumulated to form an experienced pool with high benefit values, which makes the prediction results more accurate.

Step 2: due to the influence of the traffic states before and after the training, it is necessary to determine whether the action can get excellent feedback before execution. The action a is performed according to the strategy p , and the cumulative return is calculated, after the strategy is executed. The state value function expression is listed as follows:

$$V_p(s) = \sum_{s'} p_{(s)}(s, s') (R_{p(s)}(s, s') + \gamma V_p(s')), \quad (9)$$

where $V_p(s)$ represents the degree of return according to the strategy p under state s . $p(s, s')$ represents the probability of state transition. $R(s, s')$ represents the reward obtained from $s \rightarrow s'$, and γ is a function coefficient.

5.2. Reward Actions. The reward after training in the previous state s is represented by the difference in delay time. The neural network function uses s as input. $Q(a, n_features)$ is the storage table, and $n_features$ presents the number of input neurons. Therefore, the output vector dimension is $2^{n_features} = 32$. The memory storage pool structure after the reward is $[n_features, a, r, n_features]$.

During the process of predicting the future situation, inputting the current state and outputting the Q-value are studied under various possibilities with the largest one being selected. As the reward level continued to deepen, the target results gradually approach the actual situation in which the Q-value here refers to the traffic delay index.

5.3. Training Method. Based on the construction of the abovementioned state space and reward actions, we will train the datasets from Wanjiali Road to Shuangtang Road on the elevated Wanjiali Road in Changsha. The main steps of training methods are listed as follows:

Step 1: the preprocessing of traffic data and weather data (culling abnormal data, Lagrange interpolation, and normalization).

Step 2: the selection of training sets and test sets (the time interval of training sets is from 0:00 on May 17, 2019, to 24:00 on May 24, 2019. The time interval of test sets is from 0:00 on May 25, 2019, to 12:00 on May 25, 2019).

Step 3: determining the input and output of the variables and the number of network layers (the input variables are speed, delay time, travel time, temperature, and precipitation probability. The output variable is the delay index, the number of hidden neurons in the interval [4, 13], and 3 layers of network layers).

Step 4: determining the initial weights, thresholds, learning rate, activation function, and training function (the interval of initial weight and threshold is [0, 1]. The learning rate is 0.01, the activation function uses the *Sigmoid* function, and the training function uses *Adam*).

Step 5: training the neural network model and stopping the network training when the feedback reaches the optimal state of the *Q*-value table. If it is not satisfied, modification and adjustment of the parameter values are required (learning rate and training function).

Step 6: adjusting the parameter to achieve the best prediction results which could be obtained from the prediction and input test set data.

Step 7: analyzing the prediction results to get the final experimental results.

In this paper, the LSTM forgetting, input, and output threshold activation functions are all *Sigmoid* functions. The return interval [0, 1] is consistent with human thinking. The pseudocode to build a deep recurrent Q-Learning network is shown in Algorithm 1.

6. Case Analysis

6.1. Data Description. This paper selected a part of the arterial road in Changsha, starting from Wanjiali Road to Shuangtang Road from north to south, as the research case. A crawler script written in Python 3.7 was used to capture the real-time traffic information from the big data platform of Gaode Map. The data were collected from 0:00 on May 17, 2019, to 12:00 on May 25, 2019, with a 5-min sampling interval. The collected data types include actual time, speed, delay time, travel time, temperature, probability of precipitation, and delay index. The data set sample is shown in Table 1.

The data of this case is divided into training sets and test sets after preprocessing. The time interval of the training sets

is from 0:00 on May 17, 2019, to 24:00 on May 24, 2019, and the time interval of the test sets is from 0:00 to 12:00 on May 15, 2019.

6.2. Data Preprocessing. Data preprocessing includes three steps: culling abnormal data, Lagrange interpolation, and normalization. The detailed information is shown in Figure 5.

The first step is to cull abnormal data. The abnormal data mentioned in this step refers to the data that deviates significantly from the normal interval. By deleting such kind of data, the experimental data are more realistic and the analysis results are more reasonable. Some samples of abnormal data are shown in Table 2.

The second step is the Lagrange interpolation. Lagrange interpolation is used to fill in some missing data based on the neighboring traffic datasets to improve the value of the data. This step is used to achieve data integrity and rationality.

The data filling function for this step is listed as follows:

$$L(x) = \sum_{i=0}^n y_i \prod_{j=0, j \neq i}^n \frac{x - x_j}{x_i - x_j}, \quad (10)$$

where y_i is the polynomial of degree $i-1$ and x_i is the parameter corresponding to the point i .

The third step is data normalization. The purpose of this step is to control the magnitude of the data within a small fluctuation range, reduce the impact between the magnitudes of the horizontal data, and improve prediction accuracy. The function is listed as follows:

$$x' = \frac{x - \min}{\max - \min}, \quad (11)$$

where max is the maximum value of the sample data and min is the minimum value of the sample data.

The preprocessed data are transformed into a list to form the matrix and, finally, transformed into a three-dimensional space. The three-dimensional space serves as the input of the LSTM unit to form the basic unit of the hidden layer. Every 15 rows of valid data are used as a training set and are continuously trained in 100 times. The test sets are predicted based on the training memory to obtain the prediction results.

6.3. Prediction Module Construction. According to data analysis, visualization, and platform requirements, this paper introduces NumPy, Pandas, and Matplotlib as analysis tools. Tensorflow is used as an open-source database for deep learning to build a basic library. According to the needs of the model, a variety of modules is constructed including the traffic environment module, deep reinforcement learning module, memory module, behaviour selection module, neural network module, training main program module, loss curve module, and visualization module.

The first step is to initialize the traffic data, network environment, and training parameters to build a neural network for prediction. The second step is to input training sets and test sets to the input layer. Multidimensional data

- (1) Initial network structure, the parameter is q . Initial target network, parameter $q' = q$.
- (2) Initial trials greedy parameters epsilon, learning rate, reward, attenuation coefficient gamma, number of iterations episodes. Each episode iteration round number T , training batch size, and neural network parameter rotation cycle transfer_cycle.
- (3) for an episode in Episodes do
- (4) Initial traffic state $s_t = s_0$
- (5) For t from 0 to T :
- (6) Selection behavior. (Output an integer with a range of 0 to $2^{n_{\text{features}}-1}$): Select $a_t = \text{argmax}_a Q(s, a, \theta)$ with a probability of $1-\text{epsilon}$, and randomly select the behavior a_t with a probability of epsilon.
- (7) After the behavior is determined, find all states s_{all} in the data table that match this behavior, and then randomly select one from s_{all} as s_{t+1} (If no match is found in s_{all} , the behavior is redetermined).
- (8) Put experience (s_t, a_t, r_t, s_{t+1}) into the memory pool.
- (9) Take out batch size data randomly and calculate q_{eval} and q_{next} respectively.
- (10) Construct: $y = r_1 + g\text{max}_{t+1} Q(s_{t+1}, a | q) \rightarrow q_{\text{target}}$
- (11) According to q_{eval} and q_{target} , back propagation to improve the network q .
- (12) If the number of iterations is an integer multiple of transfer_cycle, then updates $q' = q$.
- (13) Current state = s_{t+1} .
- (14) When the maximum iteration number T of a single round game is reached, the training of this round is stopped, and the traffic state is returned to the initial trial.
- (15) end for
- (16) end for

ALGORITHM 1: Algorithm pseudocode.

TABLE 1: Example of traffic state parameters returned from the Gaode big data platform.

Time frame	Speed (km/h)	Travel time (min)	Delay time (min)	Temperature (°C)	Precipitation	Delay index
2019/5/17 3:23	61	16.2	2.1	18	0.2	1.5

Note. The road delay index is used as an evaluation index for the urban congestion degree, which refers to the ratio of actual travel time to the free-flow travel time for the city residents.

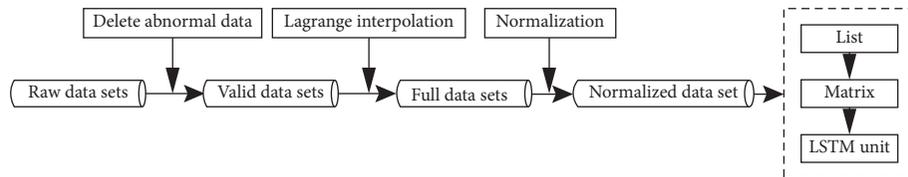


FIGURE 5: Data preprocessing flowchart.

TABLE 2: Example of abnormal data.

Time frame	Speed (km/h)	Travel time (min)	Delay time (min)	Temperature (°C)	Precipitation	Delay index
2019/5/18 17:44	55	14.3	2.1	50	0.2	1.5
2019/5/18 20:09	115	10.2	3	20	0.15	1

introduction is performed in the hidden layer, and data prediction is performed based on the experience pool in the output layer. Meanwhile, the structural dimensions of the input and output are displayed at each stage. The detailed flowchart is shown in Figure 6.

6.4. Parameter Impact Analysis. For the traffic prediction, the most critical indicators are prediction efficiency and accuracy. Therefore, the parameter impact analysis, the optimization index analysis, and the accuracy analysis are performed in the following section. The parameter impact analysis and optimization index analysis are used for the

evaluation of prediction efficiency, and the accuracy analysis is utilized for the evaluation of prediction accuracy.

During the neural network of traffic prediction, the key parameters affecting the efficiency of the experiment are studied including learning rate, reward decay, greedy, memory size, replacement interval, and batch size. The batch size is the fixed parameter, and the remaining items are variable parameters. Group 6 is the initial parameter group for comparison and analysis with other groups. Therefore, the parameter groups are divided into weakened state parameter groups (group 1–5) and strengthened state parameter groups (group 7–11).

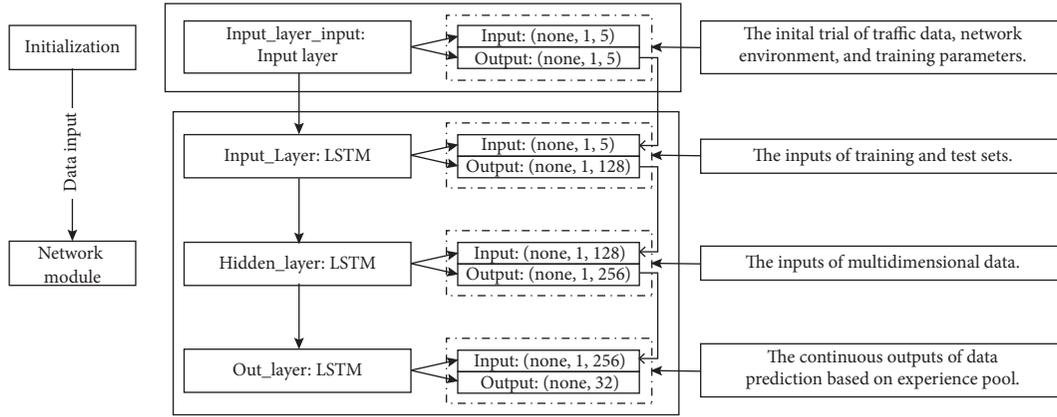


FIGURE 6: Model execution flowchart.

Each group only weakens or strengthens one parameter for comparison with group 6. To improve the discrimination of the experimental results, the selection of parameter values has an obvious gradient with the specific values which is shown in Table 3.

In order to reflect the differences in the experimental results of each group, this paper selected the indicators with obvious discrimination in the experimental results for analysis. They are the highest loss index, the lowest loss index, the maximum volatility, training time, prediction time, and total time. The detailed index distribution of each group is shown in Table 4.

The highest loss index and maximum volatility of group 4 have serious deviations from other groups and have exceeded the normal fluctuation range. It shows that the experimental efficiency analysis in group 4 has no research value. Therefore, group 4 is eliminated before performing a comparative analysis.

The qualitative analysis is performed first. Since the highest loss index, the lowest loss index, the largest volatility rate, and the total time are important parameters of experimental efficiency, visualization is performed, as shown in Figure 7.

From this, the following conclusions are reached:

- (1) The indicators of group 6 are all at the highest level, so the gradient weakening or gradient strengthening of the parameters can optimize the experimental results, but pay attention to the combination of extreme parameters. The prediction result at this time does not have actual value, such as group 4.
- (2) With the gradient adjustment of parameters, all indicators can fluctuate within a relatively small range without a sharp rise or a sharp decline. Therefore, the stability of the model proposed in this paper is confirmed.
- (3) The weakened and strengthened states of memory size and replacement interval have only slight fluctuations compared to the initial state, indicating that these two types of parameters have a little effect on the experimental efficiency.

The quantitative analysis is performed next. According to the optimization degree of each parameter, the effect of improving experiment efficiency is determined. The parameter optimization degree function is shown in the following equation:

$$O = \sum_{i=1}^4 \frac{x_0 - x_i}{x_0} \times w_i, \quad (12)$$

where O represents the optimization degree of each group, x_0 represents the parameter value of the initial group, x_i represents the parameter value of the variable group, and w_i represents the optimization weight of the corresponding parameter.

From the perspective of forecasting efficiency, the maximum volatility and total time are representative, followed by the highest loss index and the lowest loss index. Therefore, the initial weight distribution of each parameter is shown in Table 5.

Based on equation (12) and weight distribution, this paper performed optimization calculations for the weakened group compared to the initial group and the strengthened group compared to the initial group.

The following conclusions are drawn based on the quantitative results in Table 6.

- (1) The optimization effect of the groups is adjusted by the memory size and replacement interval is weaker than other groups, which further confirms the conclusion (3) in the qualitative analysis.
- (2) The optimization effects of all parameters on the experimental efficiency are quite different which indicated that there is parameter emphasis.
- (3) The weakened or strengthened groups improve the experimental efficiency, indicating that group 6 is already at or near the worst parameter combination. From this, the lowest limit of the parameter combination can be determined.

This section analyses the effect of five parameters on the experimental efficiency. The results show that memory size and replacement interval have a small effect on experimental

TABLE 3: Gradient parameter value table.

States	Group	Variable parameters					Fixed parameter	
		Learning rate	Reward decay	Greedy	Replacement	Memory size	Batch size	
Weakened states	1	0.03	0.6	0.6	200	300	32	
	2	0.03	0.6	0.6	100	400	32	
	3	0.03	0.6	0.3	200	400	32	
	4	0.03	0.3	0.6	200	400	32	
	5	0.01	0.6	0.6	200	400	32	
Initial state	6	0.03	0.6	0.6	200	400	32	
Strengthened states	7	0.05	0.6	0.6	200	400	32	
	8	0.03	0.9	0.6	200	400	32	
	9	0.03	0.6	0.9	200	400	32	
	10	0.03	0.6	0.6	300	400	32	
	11	0.03	0.6	0.6	200	500	32	

TABLE 4: Experimental result table.

Group	Highest loss index	Lowest loss index	Maximum volatility	Training time (s)	Prediction time (s)	Total time (s)
1	0.240	0.082	0.0400	1080	100	1180
2	0.230	0.066	0.0385	1088	102	1190
3	0.195	0.070	0.0375	950	92	1042
4	0.970	0.096	0.4350	1086	101	1187
5	0.154	0.070	0.0435	1096	102	1198
6	0.250	0.080	0.0600	1139	105	1244
7	0.186	0.062	0.0510	1000	93	1093
8	0.195	0.080	0.0400	1033	96	1129
9	0.157	0.065	0.0440	1046	99	1145
10	0.238	0.070	0.0400	1013	94	1107
11	0.235	0.081	0.0440	1024	96	1120

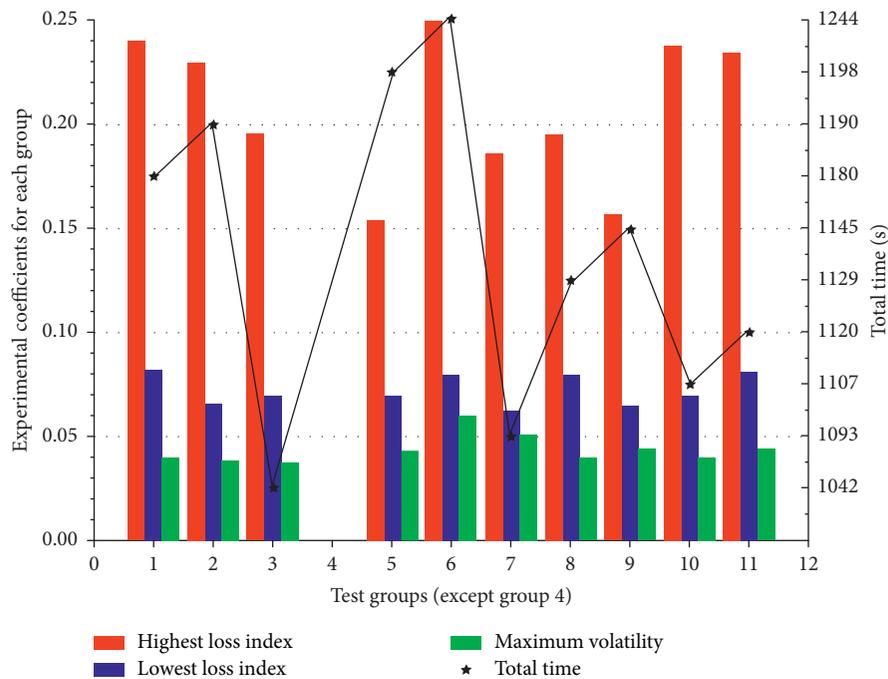


FIGURE 7: Comparison chart of highest loss index, lowest loss index, and maximum volatility.

TABLE 5: Parameter weight distribution table.

Parameter	Highest loss index	Lowest loss index	Maximum volatility	Total time
Weights	20%	20%	30%	30%

TABLE 6: The optimization situation table (except group 4).

The weakened group compared to group 6				The strengthened group compared to group 6				
1	2	3	5	7	8	9	10	11
0.1184	0.1215	0.2302	0.1954	0.1776	0.1717	0.2158	0.1276	0.1194

efficiency, while learning rate, reward decay, and greedy have a significant degree of tendency to experimental efficiency. Therefore, the three parameters are analysed for efficiency in the next section.

6.5. Optimization Index Analysis. In this section, we performed optimization index analysis on learning rate, reward decay, and greedy. Based on the abovementioned analysis, the replacement interval is set to 300 and the memory size is set to 500.

This part used the orthogonal experiments of three factors and three levels for evaluation. The three factors are learning rate, reward delay, and greedy, record as $A/B/C$. The corresponding levels are $A_1/A_2/A_3$, $B_1/B_2/B_3$, and $C_1/C_2/C_3$, respectively. Fix A and B at the levels of A_1 and B_1 , and match three levels of C with $A_1B_1C_1$, $A_1B_1C_2$, and $A_1B_1C_3$. If $A_1B_1C_3$ is optimal, fix the C_3 level. Then, let A_1 and C_3 be fixed, and match two levels of B with $A_1B_2C_3$ and $A_1B_3C_3$, respectively. After the tests, if $A_1B_2C_3$ is optimal, fix two levels of B_2 and C_3 , and try two tests $A_2B_2C_3$ and $A_3B_2C_3$. If $A_3B_2C_3$ is optimal, it is the best level combination.

When the loss curve is more stable and the optimal loss coefficient is lower, the corresponding training is better. The stability of the loss curve is reflected by the amplitude of the curve fluctuation, and the curve formed by the ratio of the loss difference to the time difference can be visually seen. The optimal loss coefficient is obtained directly from the experimental results.

The parameter combinations and corresponding results for the first test are shown in Table 7 and Figure 8.

The first test adjusts the values of greedy. Greedy's weakening adjustment significantly improves the stability of prediction and the optimal loss parameters. Therefore, reasonable adjustment of greedy helps optimize the fluctuations of the network and controls the training within a reasonable range. Based on the abovementioned experimental results and analysis, it is found that the optimal combination in the first test is group c.

The parameter combinations and corresponding results for the second test are shown in Table 8 and Figure 9.

The second test evaluates the changes in the reward delay. The effect of the reward delay on the stability of the system is more significant than that of greedy, indicating that the system is more sensitive to the reward delay. Therefore, the parameter tuning of this indicator can be perfectly combined with greedy to achieve the optimal stability of the

TABLE 7: Parameter combinations for the first test.

Group	Learning rate	Reward delay	Greedy	Optimal loss coefficient
a	0.01	0.9	0.9	0.075
b	0.01	0.9	0.6	0.074
c	0.01	0.9	0.3	0.067

system with the optimal combination in the second test in group c.

Parameter combinations and corresponding results for the third test are shown in Table 9 and Figure 10.

The object of the third test tuning is the learning rate, which not only puts forward higher requirements for system stability but also realizes the optimization progress of the loss index. Therefore, the learning rate is one of the three factors which have the greatest impact on the system. Meanwhile, the optimal loss index does not decrease with the increase in the learning rate, which indicates that the system optimization has a threshold and is not negatively correlated. The optimal combination for the third test is group f.

Therefore, the best combination obtained in all the tests is group f. The following conclusions are drawn based on the analysis of experimental results:

- (1) The learning rate, greedy, and reward delay affect the stability of the system, among which the reward delay has a greater impact. The learning rate is the only effective parameter to improve the optimal loss index.
- (2) The values of the three parameters have corresponding valid intervals. When the interval is exceeded, the prediction process fluctuates sharply and affected experimental efficiency.
- (3) No extreme fluctuation occurs during the training and prediction process. Even if there are fluctuations, they are always within a reasonable range.

6.6. Accuracy Analysis. The accuracy analysis is divided into two stages: the comparison of the predicted delay index and the actual delay index and the accuracy analysis of the traditional methods and the method proposed in this paper.

The first stage: the comparison between the predicted delay index and the actual delay index. Taking group f as the

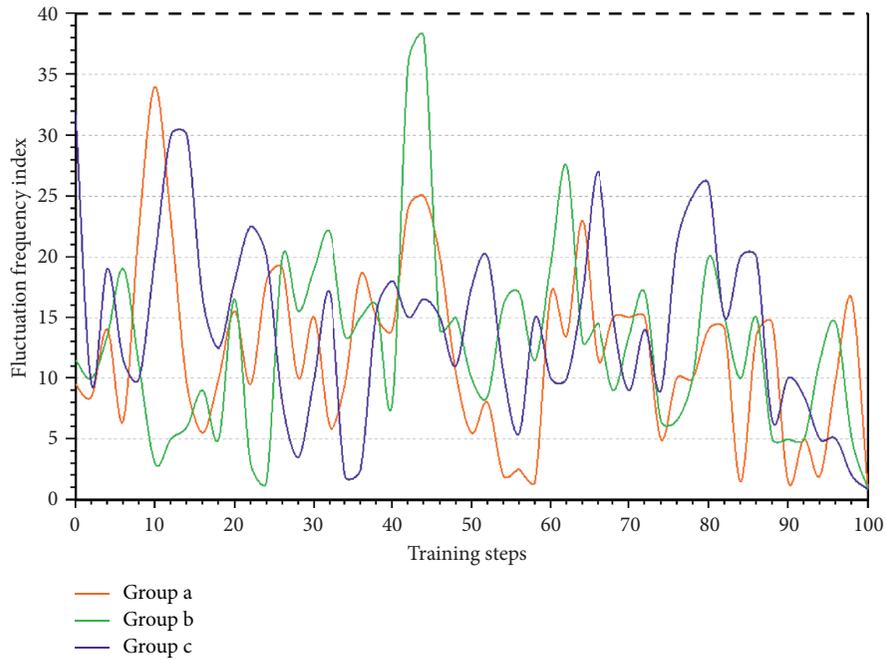


FIGURE 8: Fluctuation frequency graph of groups a, b, and c.

TABLE 8: Parameter combinations for the second test.

Group	Learning rate	Reward delay	Greedy	Optimal loss coefficient
c	0.01	0.9	0.3	0.067
d	0.01	0.6	0.3	0.069
e	0.01	0.3	0.3	0.070

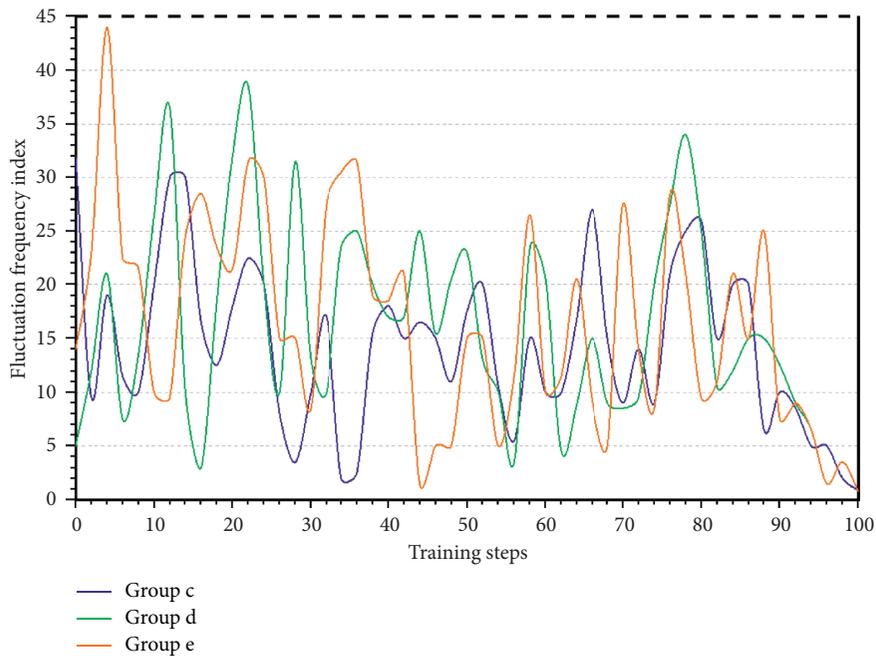


FIGURE 9: Fluctuation frequency graph of groups c, d, and e.

TABLE 9: Parameter combinations for the third.

Group	Learning rate	Reward delay	Greedy	Optimal loss coefficient
c	0.01	0.9	0.3	0.067
f	0.03	0.9	0.3	0.060
g	0.05	0.9	0.3	0.061

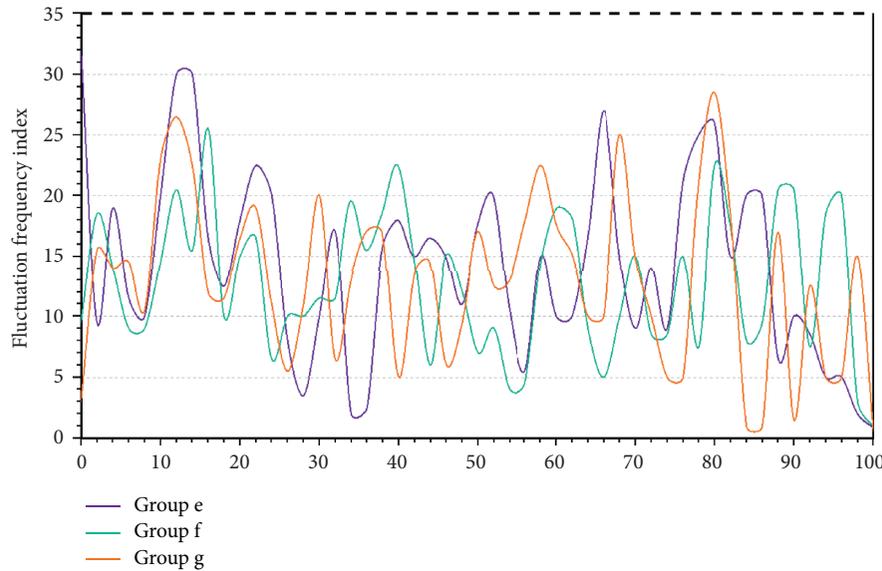


FIGURE 10: Fluctuation frequency graph of groups c, f, and g.

standard group, first draw a comparison chart based on the prediction and actual delay index, and second, calculate the prediction accuracy of groups 1–11 and A–G. This enables a preliminary accuracy evaluation.

It is known from Figure 11 that the degree of agreement between group f and the actual delay index is extremely high, and the predicted efficiency is better in the first half than in the second half. It is known from Tables 10 and 11 that group f still has the highest prediction accuracy. Therefore, the following conclusions are obtained in the first stage:

- (1) The neural network under group f is the best choice in terms of experimental efficiency and accuracy, which provides a strong guarantee for short-term traffic prediction.
- (2) There is a loss of weakness in the prediction process. If the neural network is used for long-term prediction, the network needs to be further optimized for design.
- (3) There is no absolute correlation between experimental efficiency and accuracy, so it is necessary to analyse by yourself. For example, there is a serious deviation in the experimental efficiency of group 4, and its accuracy is kept within a reasonable range.

The second stage: the accuracy analysis of the traditional method and the method mentioned in this paper. In order to further verify the superiority of the proposed method, the

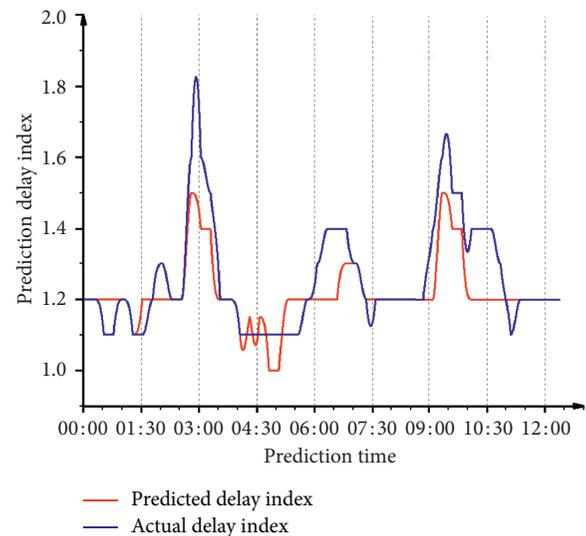


FIGURE 11: Comparison of the predicted and the actual delay index for group f.

accuracy of the proposed method is compared with the LSTM, KNN, SVM, exponential smoothing, and BP neural network. All prediction processes are based on the data used in this paper. Finally, the two representative indicators of prediction accuracy and MSE are used to measure the effectiveness of the forecast.

TABLE 10: The prediction accuracy of groups 1-11.

	1	2	3	4	5	6	7	8	9	10	11
Accuracy (%)	91.950	90.525	91.675	90.550	91.030	91.400	91.125	91.075	90.638	90.875	90.200

TABLE 11: The prediction accuracy of groups a-g.

	a	b	c	d	e	f	g
Accuracy (%)	91.850	91.202	89.528	91.600	91.400	94.178	91.328

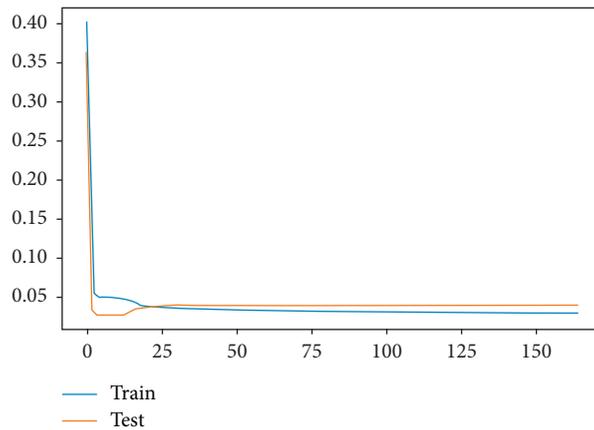


FIGURE 12: LSTM-based traffic forecast.

TABLE 12: SVM and KNN prediction accuracy table.

Error rate	SVM			KNN		
	Accuracy	Recall rate	F1	Accuracy	Recall rate	F1
0.2687	0.7292	0.7332	0.7293	0.8357	0.8409	0.8369

Note. F1 represents an indicator of the accuracy of the binary classification model. It also takes into account the accuracy rate and recall rate of the classification model.

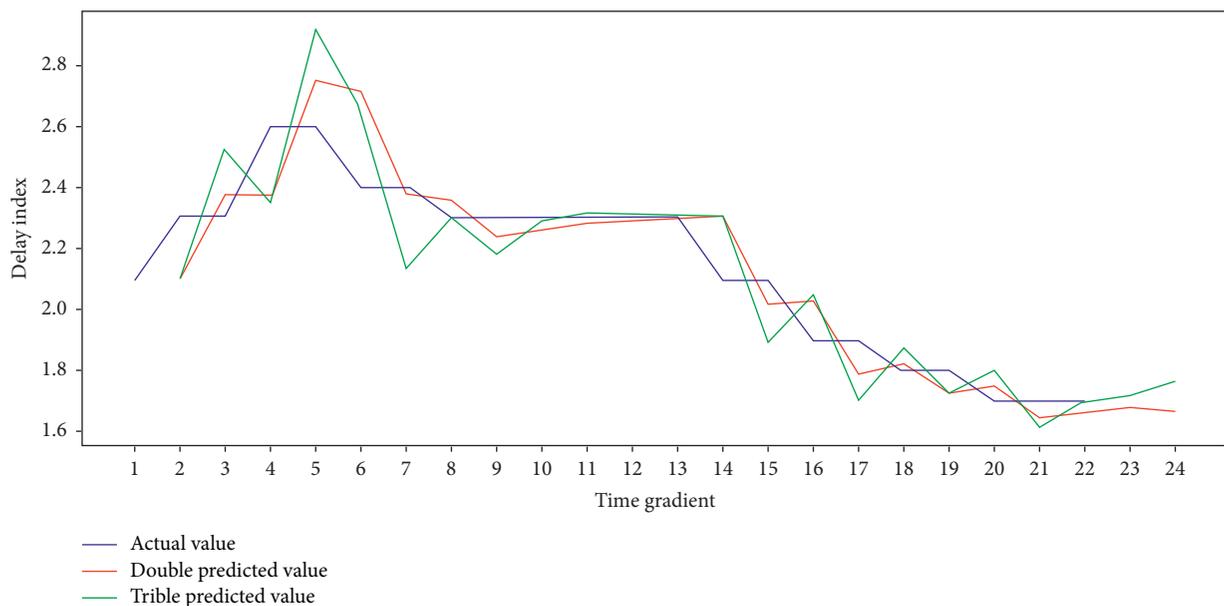


FIGURE 13: Traffic forecast based on the exponential smoothing model.

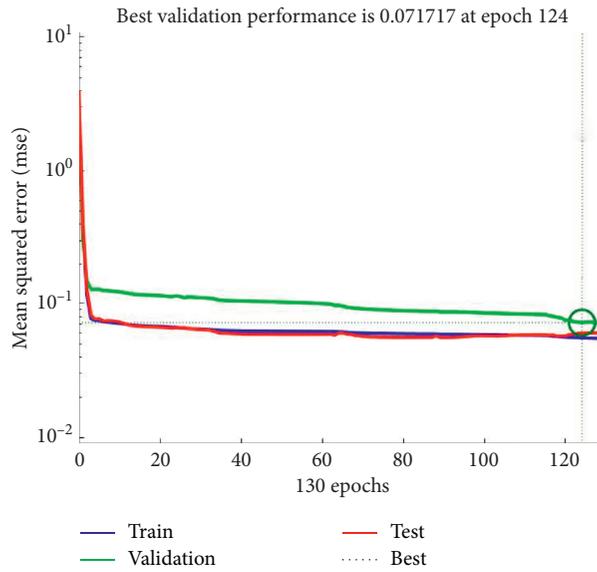


FIGURE 14: Distribution of mean squared error.

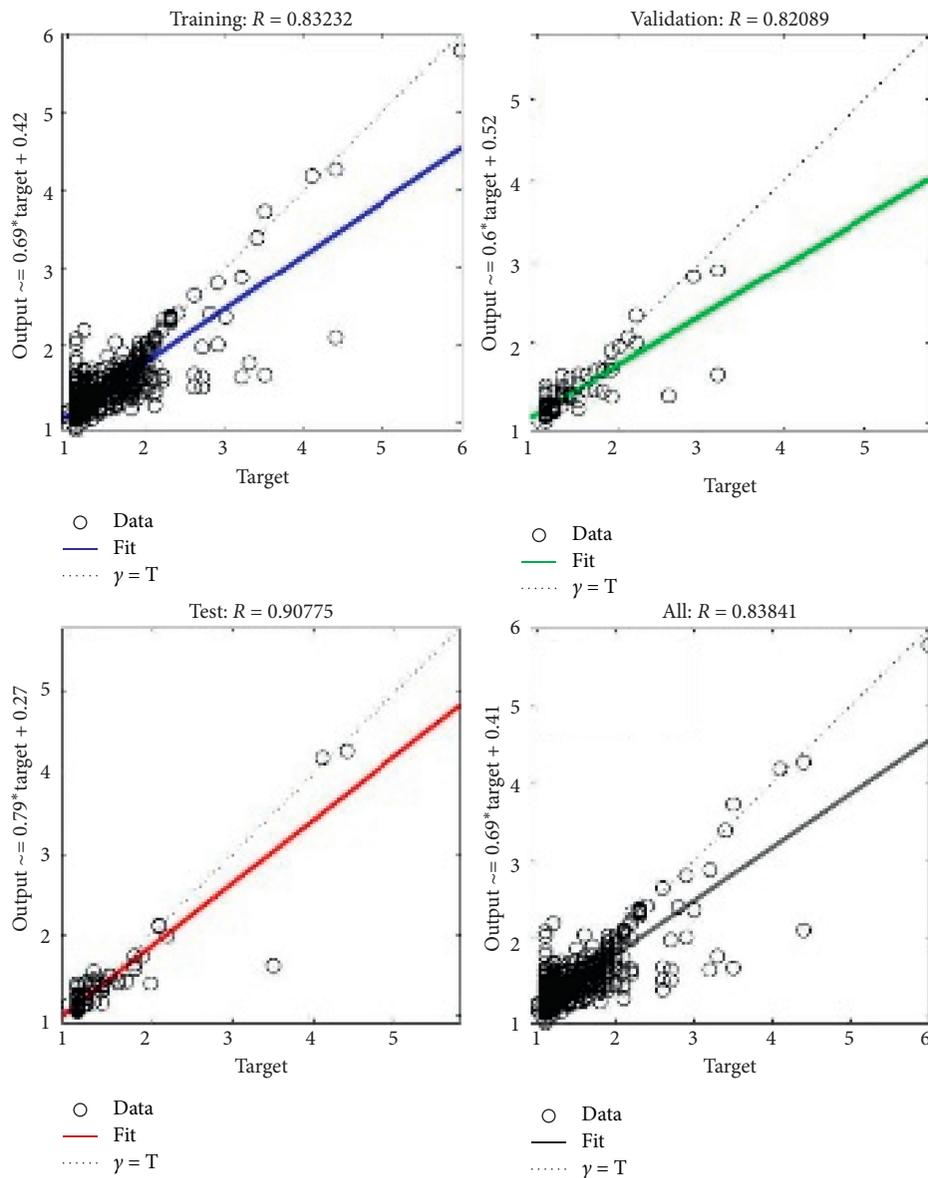


FIGURE 15: Prediction result graph.

TABLE 13: Comparison table of accuracy of all prediction methods.

	LSTM and Q-Learning	LSTM	KNN	SVM	Exponential smoothing	BP neural network
Prediction accuracy	94.178%	88.947%	83.570%	72.920%	82.898%	83.841%
MSE	0.0226	0.0595	0.0540	0.0722	0.1710	0.0717

In the LSTM, the data from 0:00 on May 17th to 24:00 on May 24th are used as the training sets, and the data from 0:00 to 12:00 on May 25th are used as the test sets. The prediction results are shown in Figure 12.

In KNN and SVM prediction, the prediction accuracy of the SVM and KNN based on data analysis is shown in Table 12.

In the exponential smoothing forecast, based on the May 20th 0:00–22 o'clock to predict the traffic state of 23–24 o'clock, the second exponential smoothing and the third exponential smoothing are performed and, finally, compared with the actual delay index. The experimental results show that the fitting curve under quadratic exponential smoothing is better than the prediction result, as shown in Figure 13.

In BP neural network prediction, speed, delay time, travel time, air temperature, and precipitation probability are used as the input matrix, and delay index is used as the output matrix. 90% are used as the training set, 5% are used as the validation set, 5% are used as the prediction set, and 10 hidden layers are used to construct the BP neural network. Training according to the Levenberg–Marquardt algorithm is performed until the best effect is achieved. Finally, the error distribution and prediction results are obtained, as shown in Figure 14 and 15.

It can be seen from Table 13 that the method proposed in this paper is more accurate than LSTM alone. Comparing the proposed method with other representative prediction methods, it is obvious that the prediction effect is better. Therefore, it further confirms the superiority of the method proposed in this paper, which can meet the demand for high efficiency and precision in traffic prediction and has the feasibility of practical application.

7. Conclusions

This paper proposed a short-term traffic flow prediction model for urban roads based on the LSTM and Q-Learning, which are used to solve the problems of low temporal correlation of traffic data, large inventory, poor comprehensive analysis, and slow feedback of prediction results. The analysis results showed that the model has excellent stability and prediction accuracy. Therefore, this model has the feasibility to apply to actual traffic scenarios and to provide accurate information guidance to reduce traffic congestion and accident rates. Moreover, this model could provide substantial method support with the development of active safety.

At the same time, the problem with this model is that the amount of data and data dimensions predicted by this training are not big enough. If there are sufficient data volume and dimensions, it will bring more mature training effects and prediction results. Therefore, the next research goal is to develop more multidimensional research directions based on deep mining of effective traffic data.

In the future, we will focus on exploring more efficient prediction methods based on the research results of this paper. Also, a series of traffic conditions such as future traffic flow, accident trends, and driving behavior trends will be predicted by introducing more relevant data.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under Grant 2019YFB1600200.

References

- [1] L. Hu, X. T. Hu, J. Wan, M. Lin, and J. Huang, "The injury epidemiology of adult riders in vehicle-two-wheeler crashes in China, Ningbo, 2011–2015," *Journal of Safety Research*, vol. 72, pp. 21–28, 2019.
- [2] Y. Y. Guo, P. Liu, Y. Wu, and F. Liu, "Traffic conflict model based on Bayesian multivariate Poisson-lognormal normal distribution," *China Journal of Highway and Transport*, vol. 31, no. 1, pp. 101–109, 2018.
- [3] Y. Guo, P. Liu, Y. Wu, and J. Chen, "Evaluating how right-turn treatments affect right-turn-on-red conflicts at signalized intersections," *Journal of Transportation Safety & Security*, vol. 12, no. 3, pp. 419–440, 2020.
- [4] Y. Guo, M. Essa, T. Sayed, M. M. Haque, and S. Washington, "A comparison between simulated and field-measured conflicts for safety assessment of signalized intersections in Australia," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 96–110, 2019.
- [5] L. Hu, J. Ou, J. Huang, Y. Chen, and D. Cao, "A review of research on traffic conflicts based on intelligent vehicles," *IEEE Access*, vol. 8, pp. 24471–24483, 2020.
- [6] H. Zhao, Y. Li, W. Hao, S. Peeta, and Y. Wang, "Evaluating the effects of switching period of communication topologies and delays on electric connected vehicles stream with car-following theory," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–11, 2020.
- [7] Y. Li, W. Chen, S. Peeta, and Y. Wang, "Platoon control of connected multi-vehicle systems under V2X communications: design and experiments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1891–1902, 2020.
- [8] J. Tang, Y. Wang, and F. Liu, "Characterizing traffic time series based on complex network theory," *Physica A: Statistical*

- Mechanics and Its Applications*, vol. 392, no. 18, pp. 4192–4201, 2013.
- [9] M. A. Quddus, “Time series count data models: an empirical application to traffic accidents,” *Accident Analysis & Prevention*, vol. 40, no. 5, pp. 1732–1741, 2008.
- [10] M. Van Der Voort, M. Dougherty, and S. Watson, “Combining Kohonen maps with arima time series models to forecast traffic flow,” *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.
- [11] B. L. Smith, B. M. Williams, and R. Keith Oswald, “Comparison of parametric and nonparametric models for traffic flow forecasting,” *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.
- [12] G. A. Davis and N. L. Nihan, “Nonparametric regression and short-term freeway traffic forecasting,” *Journal of Transportation Engineering*, vol. 117, no. 2, pp. 178–188, 1991.
- [13] X. Y. Gong and S. M. Tang, “Integrated traffic flow forecasting and traffic incident detection algorithm based on non-parametric regression,” *China Journal of Highway and Transport*, vol. 16, no. 1, pp. 83–87, 2001.
- [14] R. E. Turochy and B. D. Pierce, “Relating short-term traffic forecasting to current system state using nonparametric regression,” in *Proceedings of the 2004 IEEE Intelligent Transportation Systems Conference*, pp. 239–244, Washington, DC, USA, 2004.
- [15] J. J. Tang, Y. W. Wang, W. Hao, F. Liu, H. L. Huang, and Y. H. Wang, “A mixed path size logit-based taxi customer-search model considering spatio-temporal factors in route choice,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1347–1358, 2019.
- [16] C. Qing and W. Hao, “A methodology for measuring and monitoring congested corridors: applications in Manhattan using taxi GPS data,” *Journal of Urban Technology*, vol. 24, no. 4, pp. 1–17, 2018.
- [17] Z. W. Wang, J. Yu, W. Hao, and J. J. Tang, “Two-step coordinated optimization model of mixed demand responsive feeder transit,” *Journal of Transportation Engineering, Part A: Systems*, vol. 146, no. 3, pp. 1–12, 2020.
- [18] W. Hao, Z. Zhang, Z. Gao, K. Yi, L. Liu, and J. Wang, “Research on mandatory lane-changing behavior in highway weaving sections,” *Journal of Advanced Transportation*, vol. 2020, Article ID 3754062, 9 pages, 2020.
- [19] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, “Optimized and meta-optimized neural networks for short-term traffic flow prediction: a genetic approach,” *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 3, pp. 211–234, 2005.
- [20] X. Y. Gong and S. M. Tang, “Integrated traffic flow forecasting and traffic incident detection algorithm based on non-parametric regression,” *China Journal of Highway and Transport*, vol. 16, no. 1, pp. 82–86, 2003.
- [21] Z. S. Yang, Y. Wang, and Q. Guan, “Short-term traffic flow prediction method based on SVM,” *Journal of Jilin University (Engineering and Technology Edition)*, vol. 36, no. 6, pp. 881–884, 2006.
- [22] Z. S. Yao, C. F. Shao, and Z. H. Xiong, “Short-term traffic volumes forecasting of road network based on principal component analysis and support vector machine,” *Journal of Jilin University*, vol. 38, no. 1, pp. 48–52, 2008.
- [23] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, “Supervised weighting-online learning algorithm for short-term traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 4, pp. 1700–1707, 2013.
- [24] B. Yu, S. H. Wu, and M. H. Wang, “K-nearest neighbor model of short-term traffic flow forecast,” *Journal of Traffic and Transportation Engineering*, vol. 12, no. 2, pp. 105–111, 2012.
- [25] H. Chang, Y. Lee, B. Yoon, and S. Baek, “Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences,” *IET Intelligent Transport Systems*, vol. 6, no. 3, pp. 292–305, 2012.
- [26] H. H. Xie, X. H. Dai, and Y. Qi, “Improved K-nearest neighbor algorithm for short-term traffic flow forecasting,” *Journal of Traffic and Transportation*, vol. 14, no. 3, pp. 87–94, 2014.
- [27] Q. Bing, B. Gong, Z. Yang, Q. Shang, and X. Zhou, “Short-term traffic flow local prediction based on combined Kernel function relevance vector machine model,” *Mathematical Problems in Engineering*, vol. 2015, Article ID 154703, 9 pages, 2015.
- [28] Q. Shang, Z. S. Yang, and Z. L. Li, “Short-term traffic flow prediction based on phase space reconstruction and RELM,” *Journal of South China University of Technology (Natural Science Edition)*, vol. 44, no. 4, pp. 109–114, 2016.
- [29] B. Moghimi, A. Safikhani, C. Kamga, W. Hao, and J. Q. Ma, “Short-term prediction of signal cycle on an arterial with actuated-uncoordinated control using sparse time series models,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 8, pp. 2976–2985, 2018.
- [30] B. Moghimi, A. Safikhani, C. Kamga, and W. Hao, “Cycle-length prediction in actuated traffic-signal control using ARIMA model,” *Journal of Computing in Civil Engineering*, vol. 32, no. 2, pp. 1–10, 2018.
- [31] J. Ma, X. Li, F. Zhou, and W. Hao, “Designing optimal autonomous vehicle sharing and reservation systems: a linear programming approach,” *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 124–141, 2017.
- [32] K. Saito, M. Kimura, K. Ohara, and H. Motoda, “Efficient discovery of influential nodes for SIS models in social networks,” *Knowledge and Information Systems*, vol. 30, no. 3, pp. 613–635, 2012.
- [33] F. Wang, H. Y. Wang, K. Xu, J. H. Wu, and X. H. Jia, “Characterizing information diffusion in online social networks with linear diffusive model,” in *Proceedings of the IEEE 33rd International Conference on Distributed Computing Systems*, pp. 307–316, Philadelphia, PA, USA, 2013.
- [34] T. Kuremoto, S. Kimura, K. Kobayashi, and M. Obayashi, “Time series forecasting using a deep belief network with restricted Boltzmann machines,” *Neurocomputing*, vol. 137, pp. 47–56, 2014.
- [35] Y. Lv, Y. Duan, W. Kang, Z. X. Li, and F. Y. Wang, “Traffic flow prediction with big data: a deep learning approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [36] H. Z. Zhao, D. H. Sun, M. Zhao, and S. L. Cheng, “A multi-classification method of improved SVM-based information fusion for traffic parameters forecasting,” *Promet—Traffic & Transportation*, vol. 28, no. 2, pp. 17–24, 2016.
- [37] T. Djukic, J. W. C. Van Lint, and S. P. Hoogendoorn, “Application of principal component analysis to predict dynamic origin-destination matrices,” *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2283, no. 1, pp. 81–89, 2012.
- [38] N. Mitrovic, M. T. Asif, U. Rasheed, J. Dauwels, and P. Jaillet, “CUR decomposition for compression and compressed sensing of large-scale traffic data,” in *Proceedings of the 16th International IEEE Annual Conference on Intelligent Transportation Systems*, pp. 1475–1480, The Hague, Netherlands, 2013.

- [39] M. T. Asif, S. Kannan, J. Dauwels, and P. Jaillet, "Data compression techniques for urban traffic data," in *Proceedings of the IEEE Symposium on Computational Intelligence in Vehicles and Transportation Systems (CIVTS)*, pp. 44–49, Singapore, 2013.
- [40] Y. Y. Chen, Y. S. Lv, Z. J. Li, and F. Y. Wang, "Long short-term memory model for traffic congestion prediction with online open data," in *Proceedings of the IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 132–137, Rio de Janeiro, Brazil, 2016.
- [41] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.
- [42] X. Luo, D. Li, Y. Yang, and S. Zhang, "Spatiotemporal traffic flow prediction with KNN and LSTM," *Journal of Advanced Transportation*, vol. 2019, Article ID 4145353, 10 pages, 2019.
- [43] F. Chen, H. R. Peng, X. X. Ma, J. Y. Liang, W. Hao, and X. D. Pan, "Examining the safety of trucks under crosswind at bridge-tunnel section: a driving simulator study," *Tunnelling and Underground Space Technology*, vol. 92, 2019.
- [44] A. Miglani and N. Kumar, "Deep learning models for traffic flow prediction in autonomous vehicles: a review, solutions and challenges," *Vehicular Communications*, vol. 20, pp. 10–20, 2019.
- [45] B. Ji and E. J. Hong, "Deep-learning-based real-time road traffic prediction using long-term evolution access data," *Sensors*, vol. 19, no. 23, pp. 1–10, 2019.
- [46] Y. Zhang, T. Cheng, Y. B. Ren, and K. Xie, "A novel residual graph convolution deep learning model for short-term network-based traffic forecasting," *International Journal of Geographical Information Science*, vol. 34, no. 5, pp. 969–995, 2020.
- [47] M. Abdollahi, T. Khaleghi, and K. Yang, "An integrated feature learning approach using deep learning for travel time prediction," *Expert Systems with Applications*, vol. 139, pp. 1–10, 2019.