

CHUHAO ZHOU, Ph.D. candidate¹
E-mail: 201810101700@mail.scut.edu.cn

PEIQUN LIN, Ph.D.¹
(Corresponding Author)
E-mail: pqlin@scut.edu.cn

XUKUN LIN²
E-mail: linxukun@126.com

YANG CHENG, Ph.D.³
E-mail: cheng8@wisc.edu

¹ School of Civil Engineering and Transportation
South China University of Technology
Guangzhou 510641, Guangdong, China

² General Planning Department
Guangdong Provincial Department of Transportation
Guangzhou 510101, Guangdong, China

³ Wisconsin Traffic Operations and Safety Laboratory
University of Wisconsin Madison
Madison, WI 53706, USA

Traffic Engineering
Original Scientific Paper
Submitted: 19 Sep. 2020
Accepted: 12 Mar. 2021
DOI: 10.7307/ptt.v33i4.3709

A METHOD FOR TRAFFIC FLOW FORECASTING IN A LARGE-SCALE ROAD NETWORK USING MULTIFEATURES

ABSTRACT

Accurate traffic prediction on a large-scale road network is significant for traffic operations and management. In this study, we propose an equation for achieving a comprehensive and accurate prediction that effectively combines traffic data and non-traffic data. Based on that, we developed a novel prediction model, called the adaptive deep neural network (ADNN). In the ADNN, we use two long short-term memory (LSTM) networks to extract spatial-temporal characteristics and temporal characteristics, respectively. A backpropagation neural network (BPNN) is also employed to represent situations from contextual factors such as station index, forecast horizon, and weather. The experimental results show that the prediction of ADNN for different stations and different forecast horizons has high accuracy; even for one hour ahead, its performance is also satisfactory. The comparison of ADNN and several benchmark prediction models also indicates the robustness of the ADNN.

KEYWORDS

traffic flow prediction; deep learning; multistep prediction; toll station management.

1. INTRODUCTION

Comprehensive and accurate traffic prediction of a large-scale road network plays an important role in traffic operations and management. The road network should be considered as a whole since the

congestion of a certain road is associated with many other road sections. An accurate prediction for a large-scale road network is helpful for congestion analysis and vehicle route planning, which can help alleviate congestion through load balancing [1].

In earlier years, statistical methods were often employed for traffic prediction [2-4]. In later years, machine learning became more and more popular because of its powerful capacity for data fitting [5-7]. Due to the availability of traffic big data [8, 9] and the rapid development of neural networks [10-12], using spatial-temporal data for traffic prediction has become more effective [13].

There are a lot of studies focusing on traffic prediction in the very short-term (one step ahead) but use different methods or consider different inputs to acquire high accuracy. For example, Ryu et al. [14] used mutual information (MI) to construct the traffic state vector which used the K-Nearest Neighbor (KNN) model. It used MI to evaluate the spatial-temporal correlations between the road sections which made KNN easy to search for a similar pattern. Cheng et al. [15] used the maximum Lyapunov exponent to identify multiple sources and Bayesian estimation theory to fuse multiple measures, and the support vector regression (SVR) model was employed for prediction. Qu et al. [16] employed a six-layer deep neural network to

predict daily long-term traffic flow. It also inputs the information, including time-of-day, and holidays, to assist the model in forecasting accurately. Although they can obtain a good performance, they use too many inputs or consider a complex method just for one step. And they only predicted 5 minutes or 10 minutes ahead, which is not practical because it is insufficient for the transportation management agencies to take corresponding measures. To overcome the shortcomings of the very short-term prediction, many studies propose certain methods for multistep forecasting. Zhang et al. [17] proposed a novel deep learning framework named AGC-Seq2Seq, in which spatial and temporal dependencies were modelled separately. To increase the prediction precision, it used the attention mechanism to capture the traffic pattern. Min and Wynter [18] identified the spatial relationships by accessibility between road sections to adjust the parameters of the autoregressive integrated moving average (ARIMA) model. From those studies, we can know that if we want to employ multistep forecasting, the precision will decrease, compared with one step. A sequence-to-sequence model is a practical approach for multistep prediction; however, when the number of steps is large, it is difficult to find appropriate parameters of the model to achieve satisfying performance. Multistep forecasting through an iterative approach is easy to implement, but it may produce a cumulative error that will result in a large error while it predicts too many steps ahead.

In a large-scale road network, it is necessary to keep track of the state of different areas for traffic operations and management. Many researchers consider spatial-temporal or temporal correlation and acquire a satisfying prediction, but few studies consider the relationship between different areas and predicted locations, which means their models can usually predict only a certain location and that is not practical.

To address this issue, we propose a novel deep neural network, an adaptive deep neural network (ADNN), which can achieve a multistep prediction for multiple locations in a road network with satisfying accuracy. We use LSTM to extract the feature from traffic data and BP to acquire characteristics of contextual factors, such as prediction horizon and station index. Moreover, we illustrate the extracted characteristics, through which we can understand

the difference and similarity between different situations. The contributions of this paper are as follows:

- The model for a complete traffic prediction can be used in a large-scale road network, which shows a better way to combine the traffic data and non-traffic data.
- The implementation of a novel neural network based on our equation is proposed and the experiment using field data shows its great performance; a comparison between our model and other typical prediction models is made.
- A training algorithm suitable for a large neural network is proposed to speed up the neural network training convergence.

The rest of our paper is organised as follows. Section 2 summarises other studies related to our method in traffic flow prediction. In section 3, we describe what the equation for the traffic prediction is and how the ADNN model works. Next, we implement experiments to verify the effectiveness of our method in section 4. Finally, section 5 presents the conclusions of our paper.

2. RELATED WORKS

2.1 Traffic flow prediction models based on temporal data and spatial-temporal data

Most of the methods use historical traffic flow data in previous minutes, hours, or days. In the early years, most studies employed a statistical model for predicting. Cetin and Comert [19] presented two models, ARIMA-CUSUM and ARIMA-EM, based on basic ARIMA. It points out that the performance of basic ARIMA would become worse if an accident occurred. The methods they proposed can detect the mean of traffic flow data to avoid bias. Considering the ease of implementation for the prediction model, Shekhar and Williams [20] used three common filtering methods to automatically update the parameters of ARIMA. The method can also be applied to other approaches that are parametric to achieve adaptability. However, ARIMA is more suitable for linear time series, while the traffic flow data are non-linear and unstable, so their performance decreases. Many machine learning algorithms are proposed to address this issue. SVR is a popular approach used by many studies, such as Hong et al. [21], Hong [22], Li et al. [23]. Hong et al. [21] used SVR to recognise the complex data patterns in traffic data for a better

prediction. Additionally, they implemented continuous ant colony optimisation to search for optimal parameters for SVR. Although the accuracy of the results in that paper was sound, it took considerable effort to make a better prediction for only one road and it is not practical for a large-scale road network. In recent years, the neural network has become increasingly popular. LSTM is widely employed to predict traffic flow with time series. Li and Cao [24] implemented it to predict tourism flow and acquired a fairly good performance. Ma et al. [25] used two inputs, traffic speed and volume, and acquired better performance than using one input. It shows that LSTM outperformed other algorithms in most situations. Li et al. [26] implemented a BP-LSTM model to forecast vehicle speed, using a feature extractor to acquire characteristics from traffic data, including vehicle state and geographic information system (GIS); however, the aforementioned researchers considered the temporal correlation more, which is insufficient for a large-scale road network. To extract characteristics from spatial-temporal data, convolutional neural networks (CNNs) or CNNs combined with recurrent neural networks (RNNs) have been used. Wu and Tan [27] employed CNN to mine the spatial relationships contained in traffic data and different LSTMs were used to extract temporal features from the traffic flow of the last weekday, the last day, and the current time. However, its spatial correlation only concentrates on one road and it did not consider the impact of other locations. The method presented by Yu et al. [28] is more sophisticated. It divided a traffic network into many grids and calculated average speeds of every grid over a period. Then, it implemented a CNN to capture features among all grids and used LSTM to capture temporal features. Although it combines spatial and temporal features well, it did not distinguish the relationship between other areas of the road network and the predicted area, which means it can only predict a certain area.

From these studies, we can see that they use different methods to extract characteristics from temporal data or spatial-temporal data or both, and accurately predict traffic flow, but they also have limitations. To address the research gap, we separately extract topological characteristics from spatial-temporal data and characteristics of time series from temporal data, and we use contextual factors to distinguish the relationship between different locations, which can effectively relate the road network with the predicted locations for an accurate prediction.

2.2 Interpretability of deep neural network

Although the deep neural network has a high performance in many cases, it is always a black box. A limited number of studies have illustrated features that were extracted by the model. Wang et al. [29] used LSTM to extract spatial features and Bi-LSTM to extract temporal features to predict traffic speed. It also illustrated the feature matrix before training and after training to realise what the neural network extracts from traffic data. Wu et al. [30] proposed a deep neural network based on CNN and LSTM and employed the attention model to evaluate the scores of how the input relates to the future. According to its feature reconstruction, it can be observed that some spatial-temporal features became strong when the traffic flow data is in the deep layer.

Motivated by these studies, we also illustrated the output features of our network to realise the relationship between the characteristics and the predicted results.

3. METHODOLOGY

The traffic prediction is a process to seek the relationship between traffic features and the predicted value. There are a lot of traffic features, such as global positioning system (GPS) data, historical traffic volume, etc. that can be applied for traffic prediction. Meanwhile, the prediction may be employed for different situations, such as the traffic volume on road A or road B, and half an hour in advance or one hour ahead. However, we may use the same traffic features in different situations. It is almost impossible that all dimensions from a multi-dimensional traffic feature can affect the predicted results. Therefore, we use non-traffic data to identify what situation is best for acquiring the most important information in the traffic features. The mapping can be described as follows:

$$(s_i | a_i^j) \mapsto r_j \quad (1)$$

where s_i (state value) denotes i th traffic feature we input and a_i^j (action value) denotes what we want to extract from s_i at a certain situation j . For example, when we input the same traffic flow data to predict 5 minutes and 1 hour in advance, which may have different characteristic space, we can define the prediction horizon as a parameter to identify the important dimension of the traffic feature which makes the same traffic feature point to different characteristic space. r_j denotes the key information at a certain situation j .

The state value usually has more dimensions and is more variable than the action value. If we extract the key information from their combination, the weight of state value is easy to become larger and the effect of action value will be smaller. Hence, we use a neural network N_i^s to preliminarily extract information \hat{s}_i from s_i . According to different s_i , we can use different neural networks as our feature extractors, such as LSTM, CNN or other combinations of neural networks. Then, we use another neural network N_i^a to find an appropriate mapping value

\hat{a}_i^j for a_i^j . Considering that we can input multifeatures, the function for the prediction can be defined as follows:

$$R = N_p \left(\sum_{i=1}^V w_i N_i(\hat{s}_i, \hat{a}_i^j) \right) \tag{2}$$

where N_i denotes the neural network to extract the feature in a certain situation and w_i denotes the weight of the i th feature. V denotes how many inputs we have. Finally, we train a neural network N_p to find the appropriate weights for different features for obtaining the predicted value R .

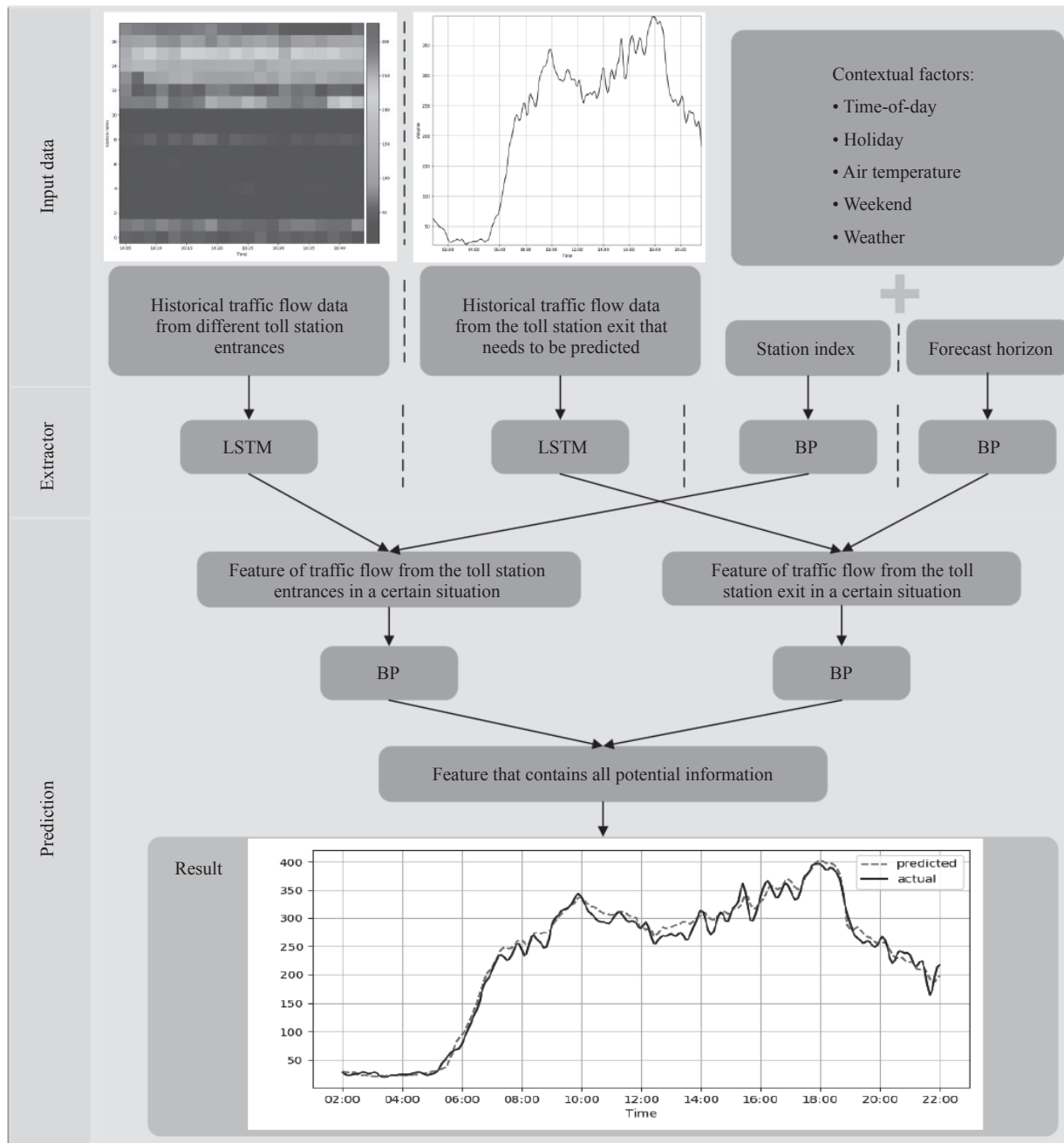


Figure 1 – The Framework of ADNN

Based on Equation 2, we propose a novel architecture of the neural network, namely, ADNN, which can seek an optimal state between completeness and precision. Figure 1 shows the framework of our neural network.

3.1 Feature extraction

To acquire the potential information of the traffic flow data, we use LSTM to extract key features in traffic flow data. Considering that the temporal relationship is more obvious in the traffic flow data of the entrance stations, we did not use CNN as our feature extractor.

In this paper, we use two LSTMs to extract features in traffic flow data of the toll station entrances and the toll station exit separately, and we need to adjust the formation of traffic flow data to adapt LSTM. The input matrices are defined as below.

$$O(T, m, n) = \begin{bmatrix} O_{T-m+1,1} & O_{T-m+1,2} & \cdots & O_{T-m+1,n} \\ O_{T-m+2,1} & O_{T-m+2,2} & \cdots & O_{T-m+2,n} \\ \vdots & \vdots & \ddots & \vdots \\ O_{T,1} & O_{T,2} & \cdots & O_{T,n} \end{bmatrix} \quad (3)$$

$$D(T, l, k) = \begin{bmatrix} d_{T-k-l+2} & d_{T-k-l+3} & \cdots & d_{T-l+1} \\ d_{T-k-l+3} & d_{T-k-l+4} & \cdots & d_{T-l+2} \\ \vdots & \vdots & \ddots & \vdots \\ d_{T-k+1} & d_{T-k+2} & \cdots & d_T \end{bmatrix} \quad (4)$$

In $O(T, m, n)$ (O_T), n represents the count of entrance stations we selected, m represents how many time points we used as input, and T represents the present time point. In $D(T, l, k)$ (D_T), T represents the same as in O_T ; however, l represents how many time steps we set and k represents the input size in one time step. We can also define the input of every time step u in O_T and r in D_T as follows:

$$u(T, m, n, i) = [o_{T-m+i,1}, \dots, o_{T-m+i,n}] \quad (i = 1, 2, \dots, m) \quad (5)$$

$$r(T, l, k, j) = [d_{T-k-l+1+j}, \dots, d_{T-l+j}] \quad (j = 1, 2, \dots, l) \quad (6)$$

Then, we input these matrices into LSTM, whose unit construction is shown in Figure 2, in which x_t could be $u(T, m, n, i)$ or $r(T, l, k, j)$, where $t=i$ or j . The equations used in LSTM cell are presented below:

$$I_{in} = \tanh(W_{in}[h_{t-1}, x_t] + b_{in}) \quad (7)$$

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (8)$$

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (9)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes I_{in} \quad (10)$$

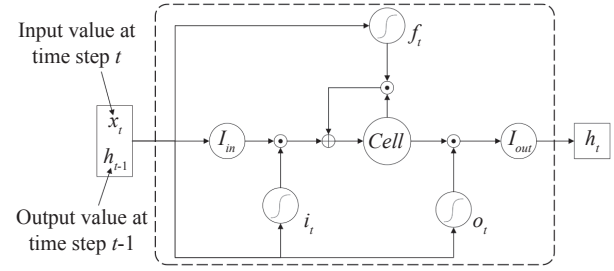


Figure 2 – The unit construction of LSTM

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (11)$$

$$I_{out} = o_t \otimes c_t \quad (12)$$

$$h_t = \tanh(W_{out}I_{out} + b_{out}) \quad (13)$$

where \otimes denotes the scalar product; $\sigma(\cdot)$ and $\tanh(\cdot)$ denote two kinds of activation functions; W_{in} , W_i , W_f , W_o , and W_{out} stand for the weight matrix of different parts in the LSTM cell; b_{in} , b_i , b_f , b_o , and b_{out} denote the bias vectors. h_t represents the output of the cell at time step t and c_t represents the cell state which represents the information contained in $u(t, m, n, i)$ at time step t . With Equations 7-13, we can acquire extracted features for prediction. The mappings can be simply defined as:

$$LSTM_O: O_T \mapsto F_O^{1 \times a} \quad (14)$$

$$LSTM_D: D_T \mapsto F_D^{1 \times b} \quad (15)$$

from the LSTM, we obtain two features F with dimensions $1 \times a$ and $1 \times b$ that contain the essential information from the entrance stations and the exit stations, respectively.

3.2 Contextual factor

Distinguished from the previous section, we need the network to know only what the situation is now, and we do not need very complex information from the contextual factors; therefore, we use a simple BP for evaluating a special vector that represents the current situation at present. The contextual factors we considered are shown in Table 1.

In addition, we also use two networks to evaluate two different vectors for matching the extracted features above. Their inputs are defined below:

$$S_1^T = [s, f_1, f_2, \dots, f_8] \quad (16)$$

$$S_2^T = [p, f_1, f_2, \dots, f_8] \quad (17)$$

Table 1 – Contextual Factors.

Factor	Notation	Description
station index	s	The index of the station that needs to be predicted.
forecast horizon	p	Which time point of the future we want to predict.
weather	f_1	The weather at the station needs to be predicted.
air temperature	f_2	The air temperature at the station.
wind	f_3	The intensity of wind at the station.
day of the week	f_4	Monday, Tuesday, ..., Sunday
weekend	f_5	Whether it is the weekend.
holiday	f_6	Whether it is a holiday.
hour	f_7	Which hour of the day.
minute	f_8	Which minute of an hour

The reason that we set s into S_1^T is that the data of entrance stations contain the information of all exit stations, so we want the network to enhance the importance of the exit station, which needs to be predicted at present. Because the historical traffic flow data of the exit station include its own developing laws, we set p into S_2^T to highlight a certain time point in the future for acquiring more precise predictions. The architecture of the neural network is presented in Figure 3. q_{in} , q_1 , q_2 , and q_{out} represent the size of the layer, and *Input* in Figure 3 could be S_1^T or S_2^T . The following shows the mappings from the contextual factor to the situation vector:

$$BP_O: S_1^T \mapsto S_O^{1 \times g} \tag{18}$$

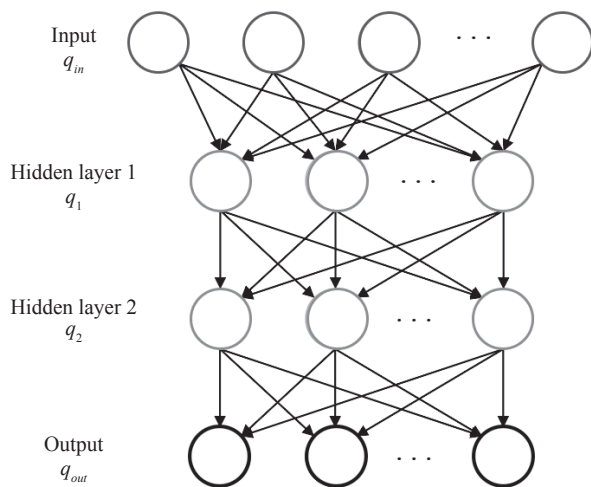


Figure 3 – The architecture of the neural network to evaluate the situation vector

$$BP_D: S_2^T \mapsto S_D^{1 \times h} \tag{19}$$

Through the BP, we obtain two situation vectors S ; the dimensions are $1 \times g$ and $1 \times h$. The vectors provide the information to the extracted features in a certain situation.

3.3 Prediction

In this section, we show how to obtain forecasting traffic volume from the situation vectors and the extracted features. First, we combine the feature and situation vectors as a whole input, and two features are combined separately. Then, through the BP neural network, we obtain the evaluation representing information of the entrance or exit, which is the most appropriate value in this situation. Finally, we input two evaluations into a new BP neural network to acquire output, which is the forecasting value we want. Figure 4 shows the process.

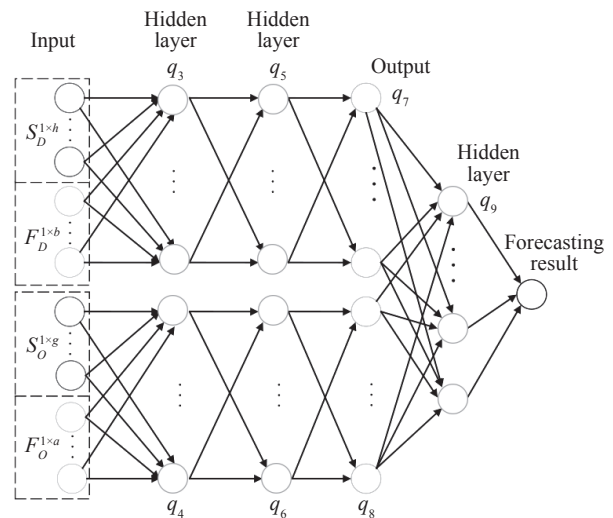


Figure 4 – Process of prediction

We defined the mappings from *Input* to *Output* in Figure 4 for a simple description in section 3, which are shown below:

$$BP_O^{predictor}: (S_O^{1 \times g}, F_O^{1 \times a}) \mapsto SF_O \tag{20}$$

$$BP_D^{predictor}: (S_D^{1 \times h}, F_D^{1 \times b}) \mapsto SF_D \tag{21}$$

3.4 Batch training

Before training the network to determine its parameters, we need to set the loss function L . The parameters are too many in our network, so we add L2 normalisation to avoid overfitting. The function is expressed below:

$$L(y, \hat{y}) = \frac{1}{N} \| (y - \hat{y}) \|_2^2 + \alpha \| w \|_2^2 \quad (22)$$

where y is the actual value and \hat{y} is the value we predicted, and N is the data count we input at one time; additionally, w represents the parameters in our network and α is a regularisation coefficient.

The network we proposed is very large if we train all data every time, and it will take a long time to converge. Hence, we adopt batch training to make our model converge, and we refer to the method in Mnih et al. [31]. Our training method can be described as follows:

Training Algorithm

Initialise training memory Q to capacity C .

Initialise ADNN with random weights w .

For epoch=1, M do

For $t=1, T_{end}$ do

Acquire O_p, D_p, S_1^t, S_2^t from the training dataset and obtain the actual value y_{ij} according to contextual factors, where i is the station index, and j is the forecast horizon.

Repeat storing input $(O_p, D_p, S_1^t, S_2^t, y_{ij})$ β_{ij} times

Sample random minibatch of inputs

$(O_w, D_w, S_1^u, S_2^u, y_{uve})$ from Q .

Obtain forecasting result \hat{y}_{uve} through inputting

(O_w, D_w, S_1^u, S_2^u) into the ADNN.

Perform a gradient descent step on $L(y_{uve}, \hat{y}_{uve})$ with respect to the ADNN parameters w .

End for

End for

In the algorithm, T_{end} represents the last time point in our training dataset, and M represents how many times we should train all data. We also add a hyperparameter β_{ij} to train the data several times. The harder the data are to converge, the bigger β_{ij} .

4. RESULTS AND DISCUSSION

To test the performance of the model we proposed, we used the actual data from toll stations in Guangdong to implement our experiments. We also compared the method in this paper and other typical prediction models, including support vector regression (SVR), k-nearest neighbour (kNN), gradient boosting decision tree (GBDT), and LSTM, in different stations and forecast horizons to evaluate the robustness of our method.

4.1 Data description

The data from the toll stations we used include 7 exit stations and 17 entrance stations from which the most vehicles drive to the exit stations. The data from the exit stations represent the number of vehicles leaving these stations and the data from the entrance stations represent the number of vehicles entering these stations. Figure 5 shows the location of the stations, where the blue markers represent entrance stations, and the red markers represent exit stations. The real traffic flow data were summarised



Figure 5 – The location of toll stations around Shenzhen, Guangdong

in a 5-minute volume, and we used the most recent actual data to predict from 5 minutes to 1 hour ahead.

We separated all data between 1 May 2018, and 31 July 2018, into training data from 1 May to 21 July and test data from the remaining data. Considering that the traffic volume at night is low, which is less important than other time periods, we forecast only the traffic volume from 6:00 a.m. to 10:00 p.m. in one day.

4.2 Training details

The hyperparameters of our network are difficult to determine, and considering the high computational cost, we did not implement a grid search. Therefore, we repeated several informal searches on the training data to find the appropriate values. The settings are shown in *Table 2*.

4.3 Forecasting results

To evaluate the performance of our method, we adopt the mean absolute percentage error (MAPE), defined as:

$$MAPE = \frac{1}{T} \sum_{i=1}^T \left| \frac{y_i - \hat{y}_i}{\hat{y}_i} \right| \tag{23}$$

where y_i represents the forecasting value of a time point in a day at a certain station, \hat{y}_i represents the actual value, and T represents the number of time points of a day.

First, we need to know how the method performs as a whole; hence, we calculate the MAPE in all days from the test data and all exit stations we employed. We can see the effect of our method in *Figure 6*.

In *Figure 6*, we can see that the method performs well in most situations, and at station 1, the MAPE is scarcely larger than 0.1. However, we also found some MAPE values approach 0.15, even 0.2. Then, we should examine why the prediction at those situations (red square) is worse than others; therefore, we selected the data at stations 1 and 2 on 22 July for predicting 20 minutes and 1 hour ahead to see the prediction of all time points and determine the reasons. The results are shown in *Figure 7*.

In *Figure 7*, we realise that the results were not as poor as we thought, but the prediction (*Figure 7d*) did not fit well either (a circle). We believe that the

Table 2 – The values of hyperparameters

Hyperparameter	Value	Description
m	18	Number of time steps in O_T
n	24	Input size of every time step in O_T
l	12	Number of time steps in D_T
k	12	Input size of every time step in D_T
number of units	64	Number of units in the LSTM
a	6	Output size of the $LSTM_O$
b	3	Output size of the $LSTM_D$
q_1, q_2	20, 9	Hidden layer size in <i>Figure 3</i>
g	2	Output size of BP_O
h	2	Output size of BP_D
q_3, q_5, q_7	18, 8, 1	The sizes of the hidden layers or output layer in <i>Figure 4</i>
q_4, q_6, q_8	23, 12, 1	
q_9	5	
N	64	Batch size
α	0.001	Regularisation coefficient
C	10000	Memory size for training
M	50	Number of times for training all data
β_{ij}	$\lceil j * 0.7 \rceil$	Number of times for repeating storing some data.
learning rate	0.005	The learning rate used by gradient descent

¹ $\lceil \cdot \rceil$ - round up

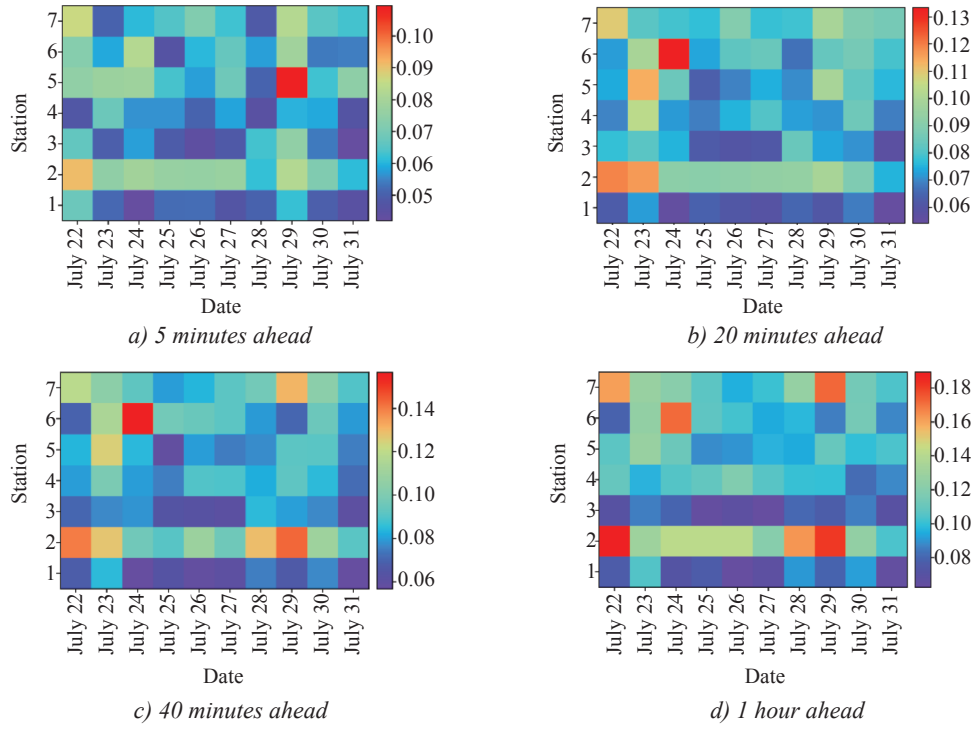


Figure 6 – MAPE in different forecast horizons

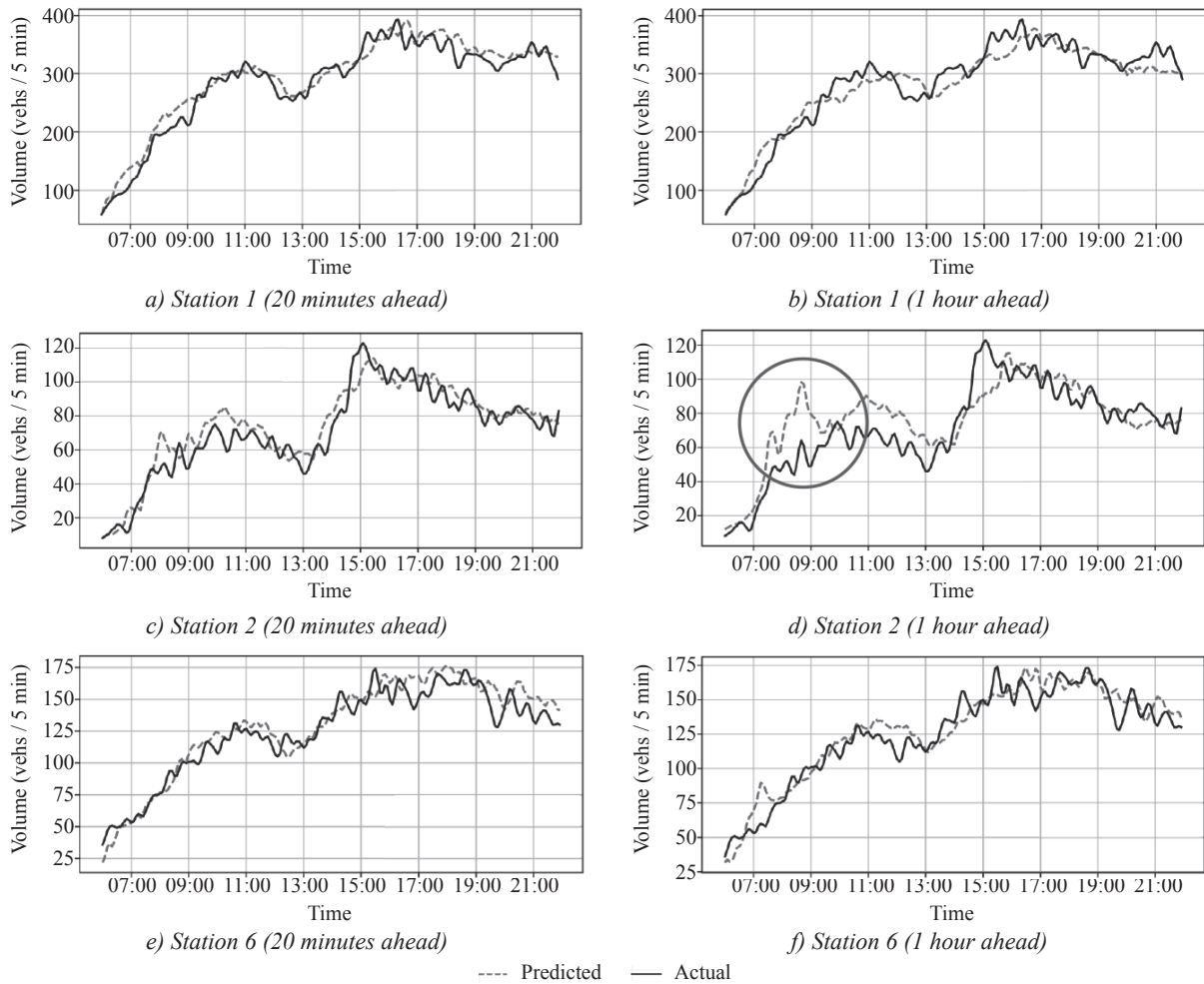


Figure 7 – Prediction at different stations and forecast horizons on July 22

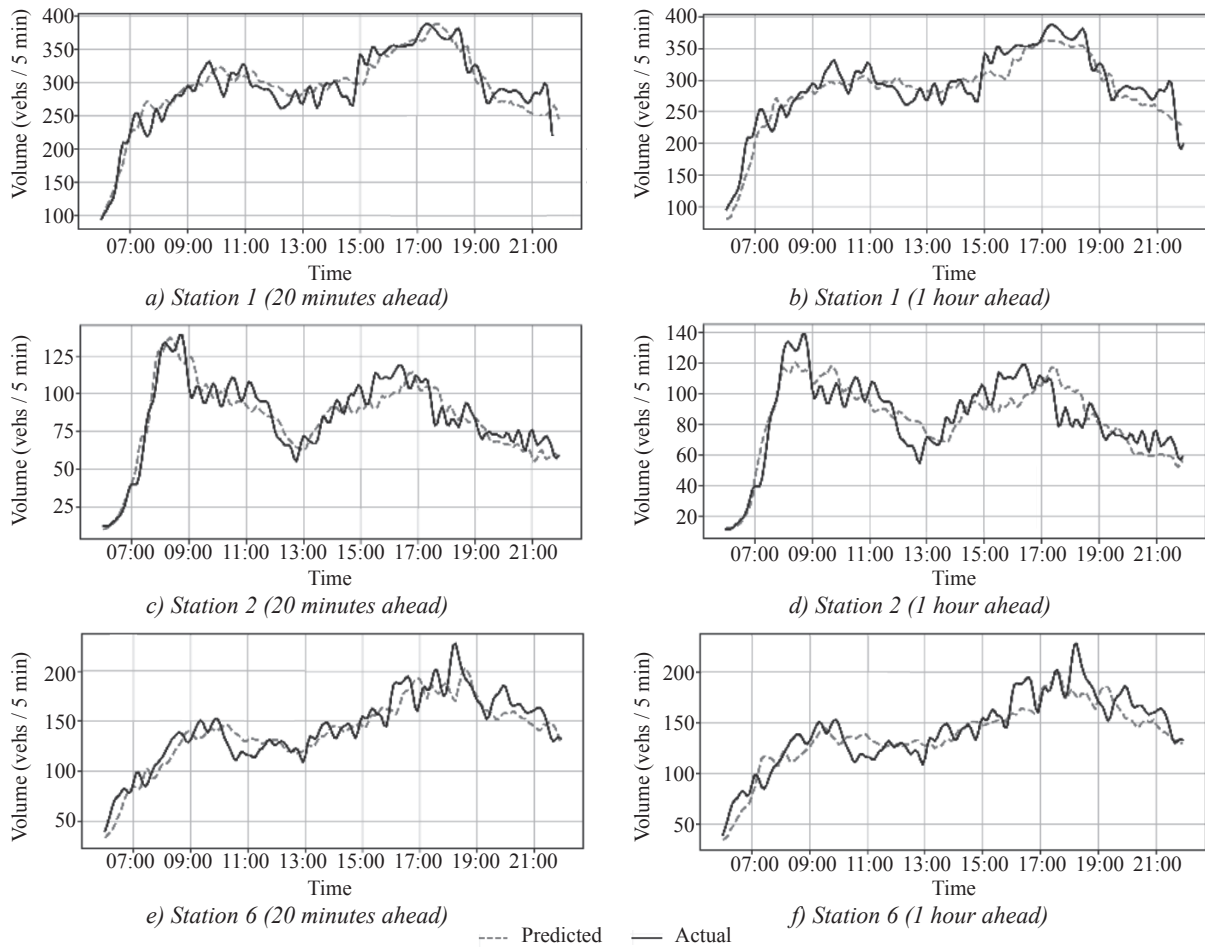


Figure 8 – Prediction at different stations and forecast horizons on July 27

magnitude of the volume is small, so the data we input is not enough to extract the information for 1 hour ahead. Compared with station 6 (Figure 7f), it had the same magnitude, and the performance of station 6 was better than station 2. Therefore, we believe that the data from station 2 on 22 July are more unstable than those from other stations. To verify that further, we selected another day to observe the prediction. Figure 8 shows the results.

These results are much better than the results shown in Figure 7. The MAPE of station 2 for 1 hour ahead decreased from 0.1895 to 0.1206. Comparing station 1 and station 2, we find that the max volumes of the two are different. It is obvious that when the traffic volume is low, it is difficult to predict too far ahead, especially at peak hours, and considering that MAPE is higher at station 2, we can see that MAPE may not be a good method for estimating the performance of our method at that station. For a better evaluation of the proposed method, we used the mean absolute error (MAE), which is expressed as follows:

$$MAE = \frac{1}{T} \sum_{i=1}^T |y_i - \hat{y}_i| \tag{24}$$

where the absolute error is not divided by the actual value; hence, we can clearly determine the difference between the actual and the predicted values. Figure 9 shows the MAE of forecasting values in the same situations as Figure 6.

The performance evaluated by MAE is completely different from MAPE. We can observe that station 1 becomes the worst. Although station 1 has the highest MAE, the error is small enough compared to the station’s max traffic volume. Because of the low fluctuation of MAE, we can also conclude that our method has good stability for application in real environments for traffic control at toll stations.

Finally, we presented MAPE and MAE of all exit stations, and the prediction is from 5 minutes to 1 hour in advance. Figure 10 shows the difference of errors at all stations under all forecast horizons.

In Figure 10, we can observe that the error basically increases as the forecast horizon increases, which corresponds to the reality that the influence of the

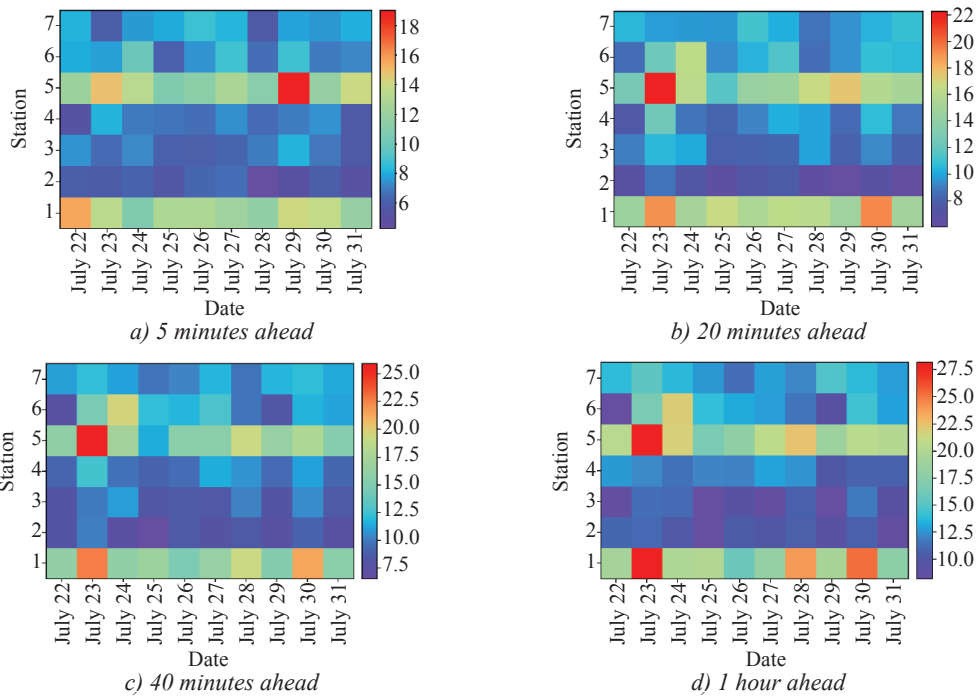


Figure 9 – MAE in different forecast horizons

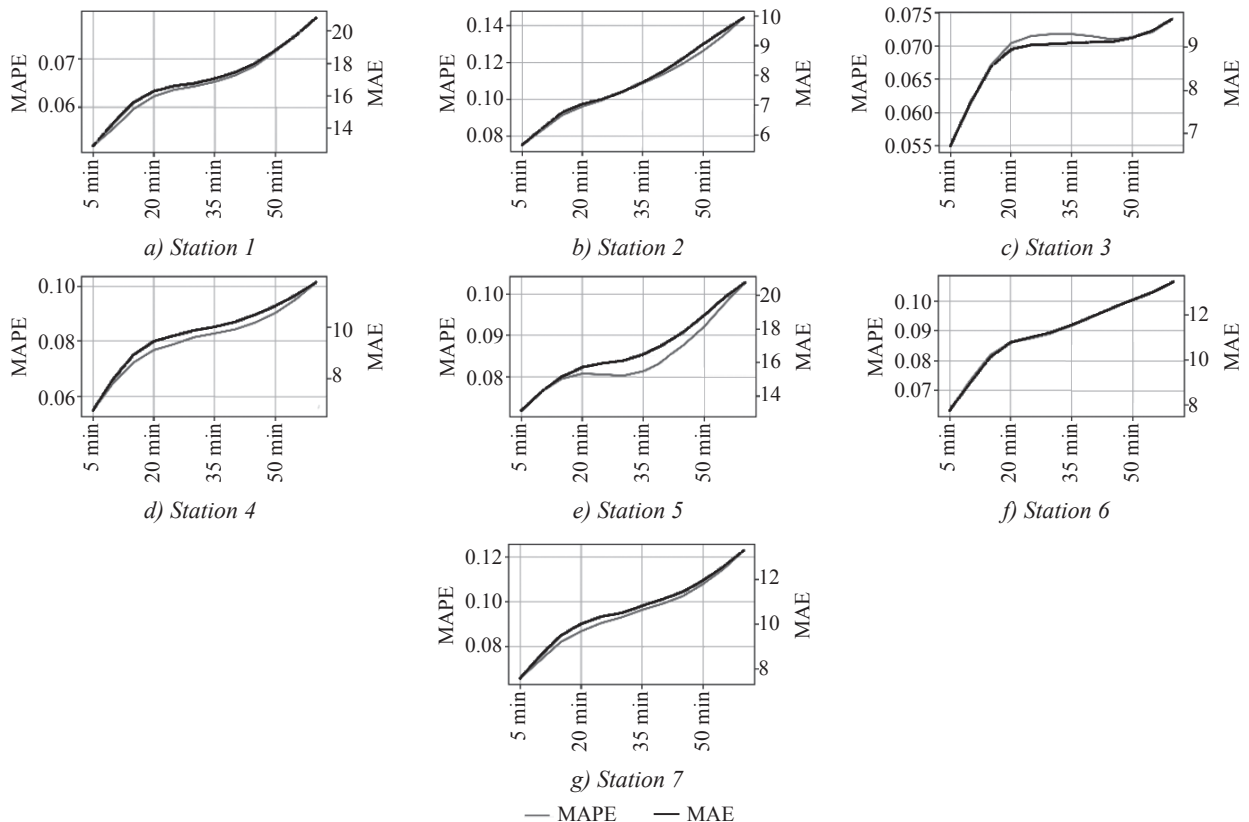


Figure 10 – Performance of all exit stations under all forecast horizons

input for predicting will weaken as the forecast horizon increases. Between 20 minutes and 35 minutes, the error rises slowly, especially at stations 3 and 5. This is because most vehicles cannot arrive in 15

minutes or over 40 minutes, which means that the influence of traffic flow at entrance stations is the highest between 20 minutes and 35 minutes, and the rest is low. Observing the MAE at stations 2 and 3,

we realised that the magnitude of traffic volume at the two stations is the same; however, the MAPE has a large difference. As mentioned above, the input to forecast the traffic flow of station 2 is not enough for predicting too far ahead.

4.4 Feature analysis

To explain why we use a BP network for mapping the contextual factors to a high dimension, we run the t-SNE [32] algorithm on the original contextual factors and the last hidden layer of the BP_O . Figure 11 shows the results.

From Figure 11, we can see that the contextual factors after mapping (left) are more distinguished. It can reflect the important information in the contextual factors. As we said, we need the contextual factors to represent the situation and make the network focus on the key information in the traffic data to acquire a more accurate prediction. Given that the complexity of the traffic data is higher than the contextual factors, if we use the original contextual factors, the network we proposed may neglect the information contained in the contextual factors and it may obtain a poor prediction.

Then, we see the differences of the extracted features (SF_O and SF_D) in our network and how the predicted results changed when we changed the contextual factors. Figure 12 shows the development of features and traffic volume by time.

In Figure 12, we presented the development of two features; the magnitudes of features and traffic volume are different because we adopted

min-max normalisation to avoid some problems that occurred in the training, which is described as follows:

$$x' = \frac{x - X_{min}}{X_{max} - X_{min}} \tag{25}$$

where x is the input at a time point, and X_{min} is the minimum value in all x , and X_{max} is the maximum value in all x .

It can be observed that the tendency of feature values is close to the traffic volume; furthermore, SF_D (green line) was closer, which means that the traffic flow data of exit stations contain more information on the development. We can see that the two features fused together can achieve a better prediction. To verify that inputting more features can obtain better performance, we compared the predictions that used one feature and two features. To better understand the effectiveness of the multifeatures, Table 3 is presented to demonstrate the MAPE of all stations with different input counts.

In Table 3, whatever the forecast horizon is and whatever the station is, using more inputs can acquire a better prediction. From Figure 13, we can also see that using two inputs is better. When we predicted not too far ahead, the MAPE was not much different. However, when we predicted 1 hour ahead, the MAPE increased substantially using one input. Since the single input we used is the traffic flow data from the exit station, it can imply that the information from the exit station is not enough for predicting too far ahead. At this time, another input of traffic flow data from the entrance station can increase the accuracy of prediction.

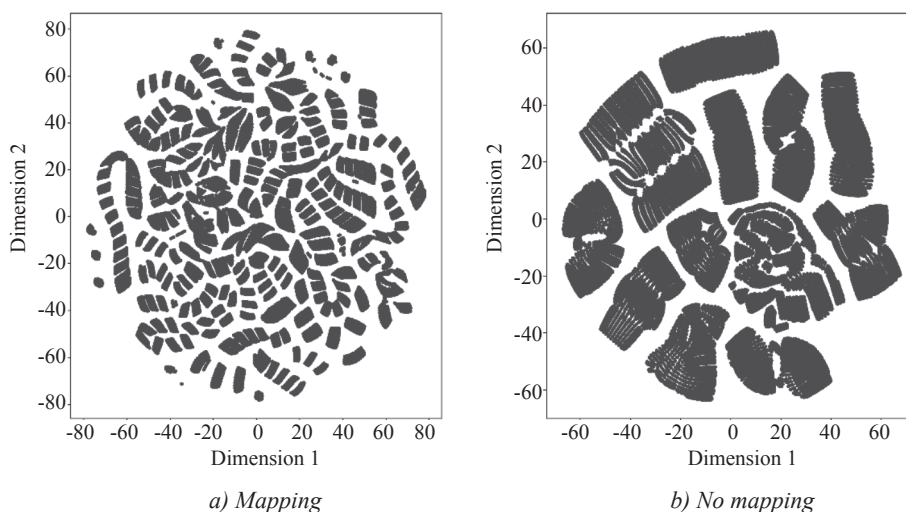


Figure 11 – Two-dimensional t-SNE embedding of the last hidden layer assigned by the BP network and the original contextual factors

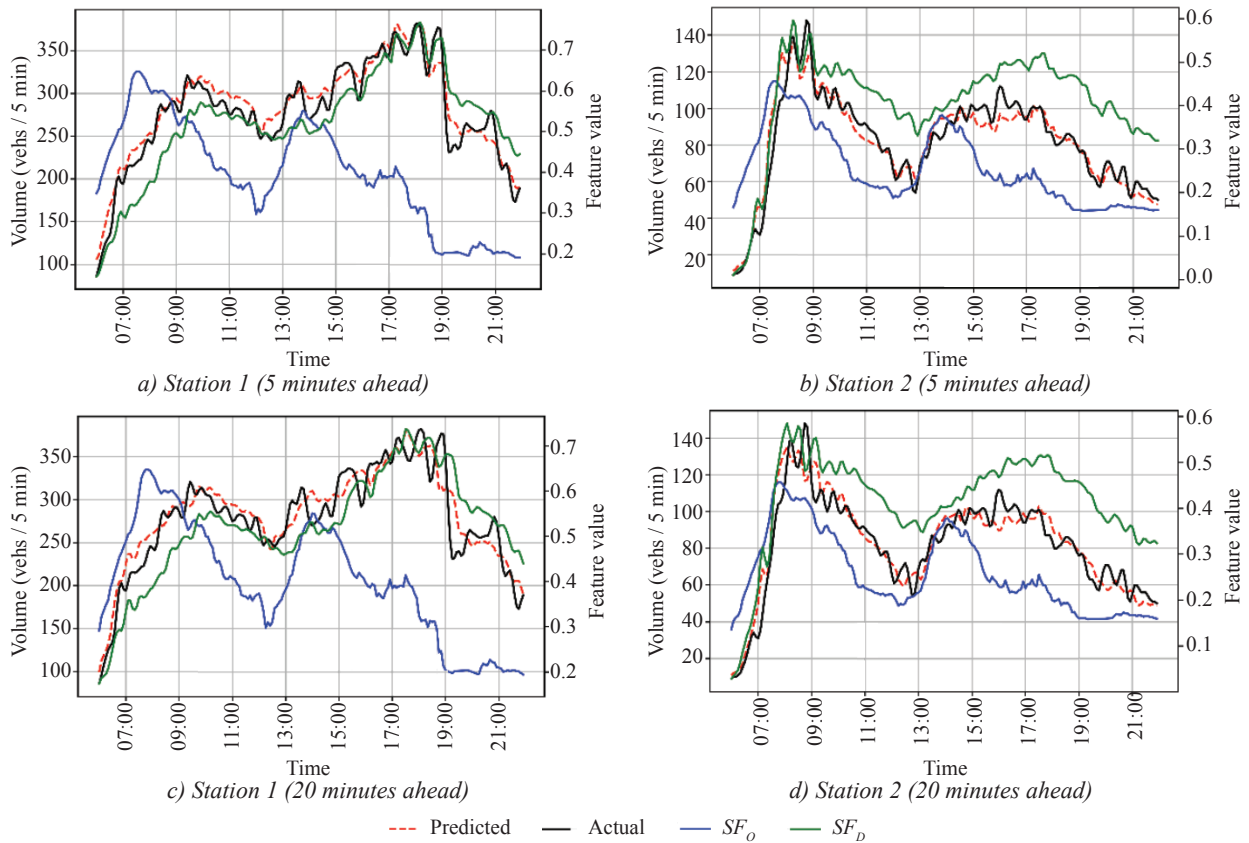


Figure 12 – The relationship between features and traffic volume (24 July)

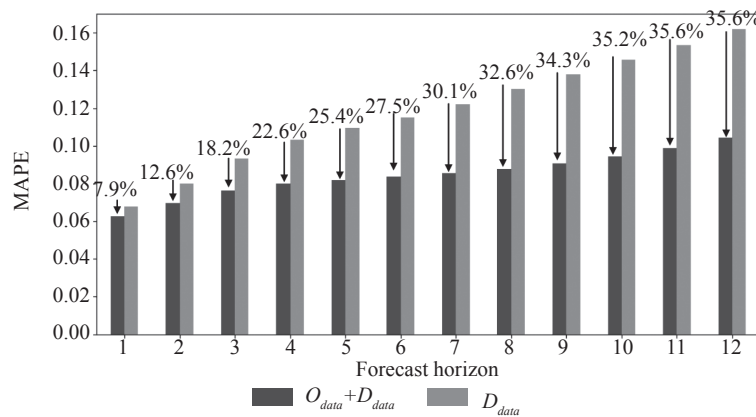


Figure 13 – MAPE in different forecast horizon over all stations with different input counts

Table 3 – MAPE of all stations in different forecast horizons with different input counts (July 30)

Station	Forecast horizon							
	5 min		20 min		40 min		1 hour	
	Input count							
	2	1	2	1	2	1	2	1
1	0.0499	0.0505	0.0685	0.0903	0.0767	0.1092	0.0918	0.1401
2	0.0716	0.0901	0.0882	0.1390	0.1105	0.1904	0.1250	0.2397
3	0.0540	0.0605	0.0709	0.0946	0.0770	0.1165	0.0860	0.1386
4	0.0587	0.0676	0.0853	0.1027	0.0846	0.1215	0.0813	0.1477
5	0.0630	0.0649	0.0827	0.0984	0.0912	0.1336	0.0995	0.1619
6	0.0540	0.0698	0.0866	0.1103	0.0947	0.1359	0.1145	0.1675
7	0.0654	0.0854	0.0885	0.1481	0.1039	0.1952	0.1139	0.2559

4.5 Comparison

To further see our method’s effectiveness, we chose four other models to compare with ours, including SVR, kNN, GBDT, LSTM. These methods are also often used for prediction and have good performance. For SVR, kNN and GBDT, considering that it is difficult to use the same input in these methods as ours, we only used the traffic flow data at the entrance stations 25 minutes before the predicted time, if there is actual data. Otherwise the latest data will be taken, because the intensity of the feature about that time point is higher than the others that were analysed above and the continuous traffic flow data at the exit station. For these methods, we also did not use the contextual factors that may be difficult to adjust for each method to become the best; instead, we decided to train one model for one situation; hence, we needed to train a total of $7 \times 12 = 84$ models on each method for all situations. For LSTM, we needed to train a total of $7 \times 1 = 7$ models for all stations, because it is suitable to predict multiple steps. The results are shown in Figure 14.

Figure 14 shows the difference between the five methods. First, other methods also had the same tendency, while the forecast horizon increased as in

ours; however, compared with others, our method was stable enough to address all situations. At station 2, the MAPE of all methods was not good, and the reason we explained before was that the intensity of the extracted features at station 2 was not enough to predict too far ahead, and the magnitude of its traffic volume was small. Hence, in this situation, not only is our method difficult to handle, but we also obtained a similar result from the other methods. Then, it can be observed that our method’s performance was not better when forecasting 5 minutes ahead or 10 minutes ahead, especially GBDT, whose MAPE was rather low, approaching 0.03, because our method considered all situations, and it decreased its precision to achieve completeness.

5. CONCLUSION

In this paper, we propose a model for traffic prediction in a large-scale road network. The proposed method can predict the traffic flow in different places and multiple steps into the future. The experiments indicate that the proposed method can predict traffic in different locations in long forecast horizons well. Compared to other methods, the proposed method had almost the same or better performance.

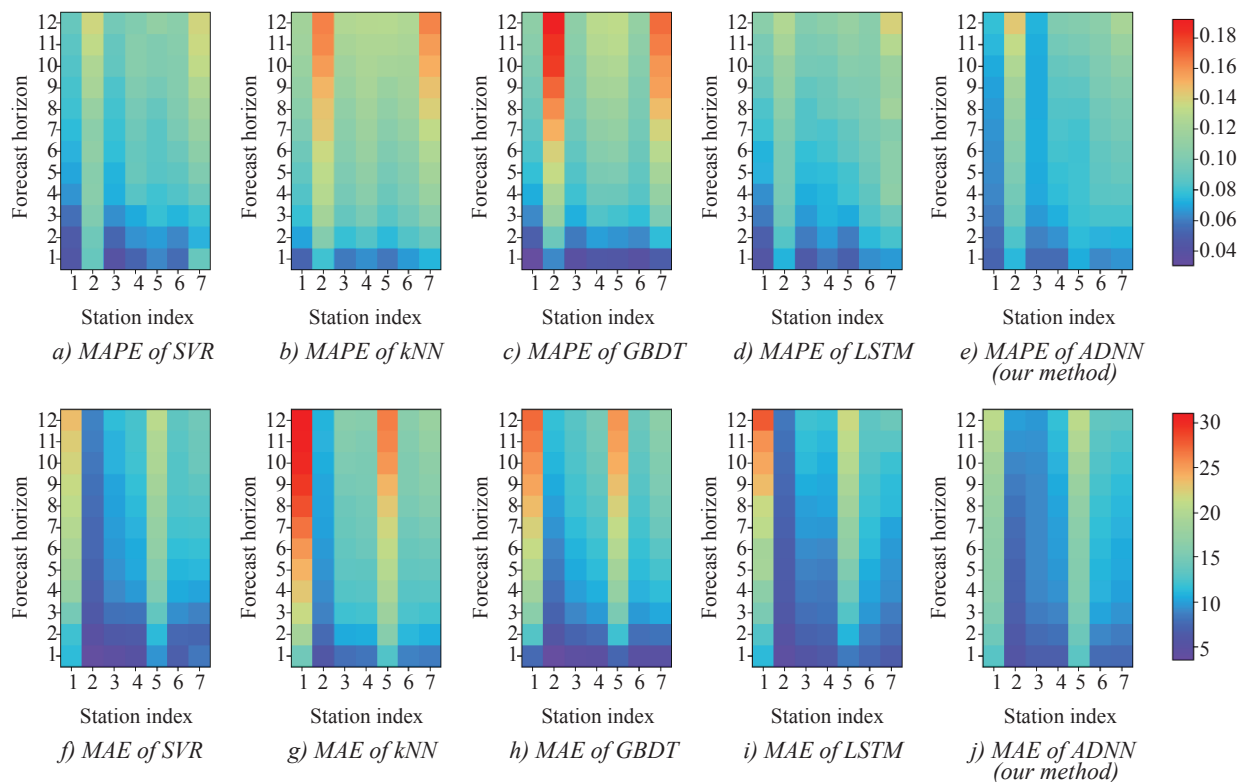


Figure 14 – MAPE and MAE calculated by different methods

We also proposed a training algorithm to speed up model convergence and achieve the global optimum for a large neural network.

Future work would include the consideration of traffic accidents, road closures, and weather impacts. For example, we could use GPS data to detect incidents. Future work would also include road network topology to compensate for data quality issues.

ACKNOWLEDGEMENTS

This research was supported by the National Natural Science Foundation of China (U1811463), the Science and Technology Program Project of Guangdong Province (2017B030307001, 2016A040404045), the Science and Technology Program Project Grant of Guangzhou (201807010008, 201803030045).

周楚昊, 林培群, 林旭坤, 程阳

一种在大规模路网下使用多特征的交通流预测方法

摘要

在大规模路网下, 准确的交通预测对于交通运营及管理有着重要作用。在该论文中, 我们提出了一个公式用于全面而准确的预测, 并且该公式体现出交通数据与非交通数据的高效结合。基于此公式, 我们构建了一种新型预测模型, 称为适应性深度神经网络 (ADNN)。在该网络中, 我们使用两个长短时记忆网络来分别提取时空特征及时间特征, 然后使用多层感知机来实现非交通参数的特征提取, 如站点、预测步长、天气。结果表明, ADNN对于不同站点以及不同步长的预测都能得到较高的准确度, 甚至对于1小时后的预测也能取得不错的效果。另外, 我们对比了其他几种预测模型, 其结果也体现出本方法的稳定性。

关键词

交通流预测; 深度学习; 多步长预测; 收费站管理

REFERENCES

- [1] Li L, et al. Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network. *Applied Soft Computing*. 2018;73: 921-932. DOI: 10.1016/j.asoc.2018.09.023
- [2] Lin W-H. A Gaussian Maximum Likelihood Formulation for Short-Term Forecasting of Traffic Flow. *2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585)*; 2001. p. 150-155. DOI: 10.1109/ITSC.2001.948646
- [3] Sun H, Liu HX, Xiao H, Ran B. Use of Local Linear Regression Model for Short-Term Traffic Forecasting. *Transportation Research Record: Journal of the Transportation Research Board*. 2003;1836(1): 143-150. DOI: 10.3141/1836-18
- [4] Yang F, Yin Z, Liu HX, Ran B. Online recursive algorithm for short-term traffic prediction. *Transportation Research Record*. 2004;1879(1): 1-8. DOI: 10.3141/1879-01
- [5] Bermolen P, Rossi D. Support vector regression for link load prediction. *Computer Networks*. 2009;53(2): 191-201. DOI: 10.1016/j.comnet.2008.09.018
- [6] Hwang S, Guevarra IF, Yu BO. Slope failure prediction using a decision tree: A case of engineered slopes in South Korea. *Engineering Geology*. 2009;104(1-2): 126-134. DOI: 10.1016/j.enggeo.2008.09.004
- [7] Juhos I, Makra L, Tóth B. Forecasting of traffic origin NO and NO₂ concentrations by Support Vector Machines and neural networks using Principal Component Analysis. *Simulation Modelling Practice and Theory*. 2008;16(9): 1488-1502. DOI: 10.1016/j.simpat.2008.08.006
- [8] Hong W, Dong Y, Zheng F, Lai C. Forecasting urban traffic flow by SVR with continuous ACO. *Applied Mathematical Modelling*. 2011;35(3): 1282-1291. DOI: 10.1016/j.apm.2010.09.005
- [9] Zhao P, Hu H. Geographical patterns of traffic congestion in growing megacities: Big data analytics from Beijing. *Cities*. 2019;92: 164-174. DOI: 10.1016/j.cities.2019.03.022
- [10] Cui Z, Henrickson K, Ke R, Wang Y. Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting. *IEEE Transactions on Intelligent Transportation Systems*. 2019;21(11): 4883-4894. DOI: 10.1109/tits.2019.2950416
- [11] Morcos AS, Barrett DGT, Rabinowitz NC, Botvinick M. On the importance of single directions for generalization. *ArXiv*. 2018. Available from: <http://arxiv.org/abs/1803.06959> [Accessed 19th March 2021].
- [12] Yang B, et al. Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*. 2019;332: 320-327. DOI: 10.1016/j.neucom.2018.12.016
- [13] Do LNN, et al. An effective spatial-temporal attention based neural network for traffic flow prediction. *Transportation Research Part C: Emerging Technologies*. 2019;108: 12-28. DOI: 10.1016/j.trc.2019.09.008
- [14] Ryu U, et al. Construction of traffic state vector using mutual information for short-term traffic flow prediction. *Transportation Research Part C: Emerging Technologies*. 2018;96: 55-71. DOI: 10.1016/j.trc.2018.09.015
- [15] Cheng A, et al. Multiple sources and multiple measures based traffic flow prediction using the chaos theory and support vector regression method. *Physica A*. 2017;466: 422-434. DOI: 10.1016/j.physa.2016.09.041
- [16] Qu L, et al. Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with Applications*. 2019;121: 304-312. DOI: 10.1016/j.eswa.2018.12.031
- [17] Zhang Z, et al. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerging Technologies*. 2019;105: 297-322. DOI: 10.1016/j.trc.2019.05.039

- [18] Min W, Wynter L. Real-time road traffic prediction with spatio-temporal correlations. *Transportation Research Part C: Emerging Technologies*. 2011;19(4): 606-616. DOI: 10.1016/j.trc.2010.10.002
- [19] Cetin M, Comert G. Short-term traffic flow prediction model with Regime Switching Models. *Transportation Research Record*. 2006;1965(1): 23-31. DOI: 10.3141/1965-03
- [20] Shekhar S, Williams BM. Adaptive seasonal time series models for forecasting short-term traffic flow. *Transportation Research Record*. 2007;2024(1): 116-125. DOI: 10.3141/2024-14
- [21] Hong W, Dong Y, Zheng F, Lai C. Forecasting urban traffic flow by SVR with continuous ACO. *Applied Mathematical Modelling*. 2011;35(3): 1282-1291. DOI: 10.1016/j.apm.2010.09.005
- [22] Hong W. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing*. 2011;74(12-13): 2096-2107. DOI: 10.1016/j.neucom.2010.12.032
- [23] Li M, Hong W, Kang H. Urban traffic flow forecasting using Gauss-SVR with cat mapping, cloud model and PSO hybrid algorithm. *Neurocomputing*. 2013;99: 230-240. DOI: 10.1016/j.neucom.2012.08.002
- [24] Li Y, Cao H. Prediction for Tourism Flow based on LSTM Neural Network. *Procedia Computer Science*. 2018;129: 277-283. DOI: 10.1016/j.procs.2018.03.076
- [25] Ma X, et al. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*. 2015;54: 187-197. DOI: 10.1016/j.trc.2015.03.014
- [26] Li Y, Chen M, Zhao W. Investigating long-term vehicle speed prediction based on BP-LSTM algorithms. *IET Intelligent Transport Systems*. 2019;13(8): 1281-1290. DOI: 10.1049/iet-its.2018.5593
- [27] Wu Y, Tan H. Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework. *ArXiv*. 2016. Available from: <http://arxiv.org/abs/1612.01022> [Accessed 19th March 2021].
- [28] Yu H, et al. Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks. *Sensors*. 2017;17(7): 1501. DOI: 10.3390/s17071501
- [29] Wang J, Chen R, He Z. Traffic speed prediction for urban transportation network: A path based deep learning approach. *Transportation Research Part C: Emerging Technologies*. 2019;100: 372-385. DOI: 10.1016/j.trc.2019.02.002
- [30] Wu Y, et al. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*. 2018;90: 166-180. DOI: 10.1016/j.trc.2018.03.001
- [31] Mnih V, et al. Human-level control through deep reinforcement learning. *Nature*. 2015;518(7540): 529-533. DOI: 10.1038/nature14236
- [32] Maaten L, Hinton G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*. 2008;9: 2579-2605. Available from: <https://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf> [Accessed 19th March 2021].