Zhixing CHEN, M.A.[1]
E-mail: ChenZhixing@cug.edu.cn
Guizhou ZHENG, Ph.D.[1]
(Corresponding author)
E-mail: zhenggz@cug.edu.cn
[1] Research Centre for Spatial Planning
and Human-Evironment System Simulation,
School of Geography and Information Engineering
China University of Geosciences
Wuhan 430074, China

Intelligent Transport Systems
Original Scientific Paper
Submitted: 13 July 2021
Accepted: 23 Dec. 2021

A BIDIRECTIONAL CONTEXT-AWARE AND MULTI-SCALE FUSION HYBRID NETWORK FOR SHORT-TERM TRAFFIC FLOW PREDICTION

ABSTRACT

Short-term traffic flow prediction is to automatically predict the traffic flow changes in a period of future time based on the extraction of the spatiotemporal features in the road network. For governments, timely and accurate traffic flow prediction is crucial to plan road management and improve traffic efficiency. Recent advances in deep learning have shown their dominance on short-term traffic flow prediction. However, previous methods based on deep learning are mainly limited to temporal features and have so far failed to predict the bidirectional contextual spatiotemporal relationship correctly. Besides, the precision and the practicality are limited by the road network scale and the single time scale. To remedy these issues, a Bidirectional Context-aware and Multi-scale fusion hybrid Network (BCM-Net) is proposed, which is a novel short-term traffic flow prediction framework to predict timely and accurate traffic flow changes. In BCM-Net, the Bidirectional Context-aware (BCM) block is added to the feature extraction structure to effectively integrate spatiotemporal features. The Interpolation Back Propagation sub-network is used to merge multi-scale information, which further improves the robustness of the model. Experiment results on diverse datasets demonstrated that the proposed method outperformed the state-of-the-art methods.

KEYWORDS

intelligent transportation system; gated recurrent unit; short-term traffic flow prediction; Bidirectional Context-aware; interpolation back propagation.

1. INTRODUCTION

A large amount of high quality traffic flow data can be obtained, providing an important data source for automatic short-term traffic flow prediction. Short-term traffic flow prediction is considered to be an important tool for Intelligent Transportation System (ITS), which is essentially a comprehensive application of traffic and service control using advanced information technology, sensor technology, artificial intelligence, etc. The system aims to strengthen the connection between vehicles, roads, and users, so as to form a new system that is timed, accurate, and efficient [1]. Short-term traffic flow prediction has been applied in diverse fields, such as urban planning, traffic supervision, traffic control, automated driving services, and geographical information upgrading.

The short-term traffic flow prediction can be divided into classical statistical method, nonlinear method, and machine learning method. The classical statistical method takes advantage of the different mathematical and statistical models, which can extract the temporal feature with low computational complexity. The nonlinear method is very suitable for forecasting under complex traffic systems, which has more complicated mathematical theoretical calculations. Compared with traditional time series forecasting methods, machine learning based on knowledge discovery, which can handle massive amounts of data, is accepted as the most straightforward and effective solution.

There are quite a few contributions in literature on short-term traffic flow prediction. Previous studies [2, 3] mainly focused on the construction of a statistical theory model, which requires a lot of manual intervention and parameter settings. Research in the field of neurology [4] has found that cat brain neurons process the information they receive in a hierarchical manner. Inspired by bionics, deep learning models [5, 6] for forecasting time series are proposed. In fact, the traditional Recurrent Neural Network (RNN) model will gradually weaken or lose its ability to learn distant information because of the gradient decline in a large traffic network. Long-Short-Term Memory (LSTM) model is proposed to solve this problem [7, 8], which realises the functions of memory, forgetting, and updating information through the structure of gates, while the model inevitably increases the amount of calculation. Moreover, traffic flow data in different Vehicle Detection Stations (VDSs) still have spatial features to mine. Studies like [9, 10] combine Convolutional Neural Network (CNN) and RNN to propose Convolutional Long-Short-Term Memory (Conv-LSTM) model. But the model ignores the bidirectional contextual relevance of time features and the internal relationship of the high-dimensional feature vectors. Research on Graph Convolution Network (GCN) resulted with the construction of the topological calculation diagram based on the road network structure [11], which makes the model difficult to generalise. It is noted that the previous studies mainly pay attention to traffic flow data on a single time scale. A multi-scale short-term traffic flow prediction model based on wavelet transform is designed in [12], while parametric model lacks the generalisation ability. However, most models use only about 10 VDSs datasets, which are not suitable for construction of ITS applications on large traffic road networks. Therefore, it is necessary to propose a prediction model that can meet the requirements of efficient calculations at multiple time scales and a large road network scale.

To solve the mentioned issues, a Bidirectional Context-Aware and Multi-Scale Network (BCM-Net) is proposed for short-term traffic flow prediction. BCM-Net uses multi-layer superimposed structure for capturing spatiotemporal features, and the Bidirectional Context-Aware (BCA) block is added to the feature extraction sub-network. To aggregate multi-scale contextual information, the feature maps obtained by the one-dimensional (1D) convolution and BCA block are passed through the Interpolation Back Propagation (IBP) sub-network. By optimising the loss function, the model prediction accuracy is further improved. Moreover, we design comparative experiments on different time scales and different road network scales to verify whether the model can solve the proposed traffic problem.

From the above, the main contributions of this paper are as follows:

1) BCM-Net is proposed to address the difficulty of short-term traffic flow prediction with multi-scale information in the large road network scale. Specifically, the framework is an end-to-end deep CNN plus RNN architecture, which integrates the low-level detail information, high-level semantic information, and bidirectional contextual information in an interweaved way.
2) To improve the feature representation, the BCA block is constructed to capture spatiotemporal contextual information. BCA block is appended on the feature extraction sub-network. This improves the feature expression of short-term traffic flow prediction.
3) By using IBP sub-network as the multi-scale merging method, the training results of the proposed network are no longer limited by the single scale information. With the optimisation allowing the model to train better, the proposed network achieved better robustness for large road network scale.

The rest of this paper is organised as follows. Section 2 describes the study area for short-term traffic flow prediction and the details of the dataset. In Section 3, the proposed BCM-Net for multi-scale traffic flow prediction is introduced. The experimental results and analysis are reported in Section 4. Finally, conclusions are presented in Sections 5.

2. RESEARCH AREA AND DATA

2.1 Research area

Figure 1 shows the traffic distribution of the 40 VDSs selected in this experiment. The experimental data were obtained from the Caltrans Performance Measurement System (PeMS), an open traffic database in California, USA. The PeMS system detects the traffic flow on the road surface in real time by a variety of sensors, including the loop coil vehicle detector, very high frequency (VHF) radar, and electromagnetic sensors, etc. After a certain process, the original sample information is aggregated into a 5-min scale and stored in the database, and then published on the Internet [13]. We can obtain data with a time scale of 5-min and derive data from it for 10-min or 15-min and so on. Moreover, the size of road network can be limited to 10, 20, 30 VDSs and so on. The top 34 VDSs from 1 January to 10

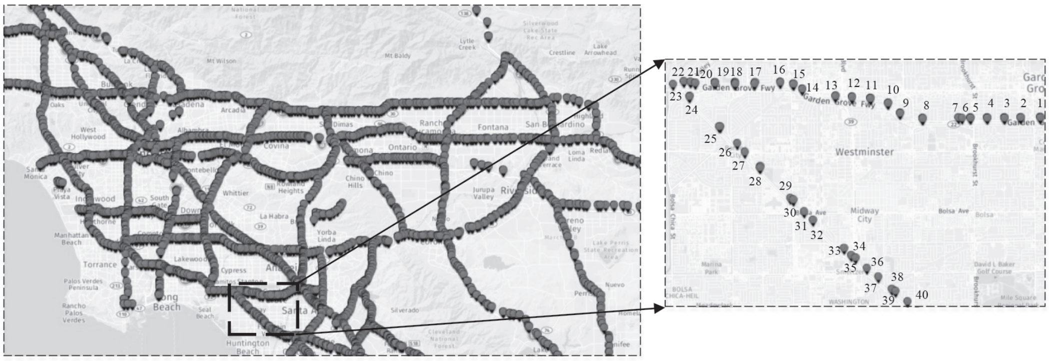


Figure 1 – Research area

July 2018 are selected as our initial date set with 31 768 samples as the training set, 6 353 samples as the test set and 8 000 samples as the verification set.

2.2 Data pre-processing

As shown in *Figure 2*, the overall trend of the traffic flow change for five consecutive Mondays was similar. The peaks, troughs, rising, and falling stages of the curve are basically coincided. The similarity and the periodicity of the traffic flow data conformed to the law of daily life and the requirement of the RNN for training data. *Figure 3* shows the traffic flow of a VDS during a week from 1 January to 7 January 2018. It shows that the traffic flow data of five working days had a similar change trend. This was caused by the fixed working mode of the fixed

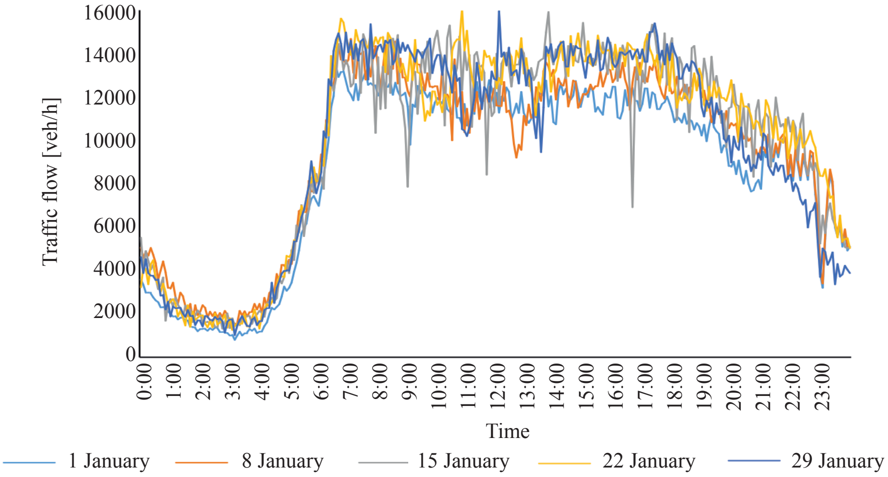


Figure 2 – Traffic flow for 5 consecutive Mondays

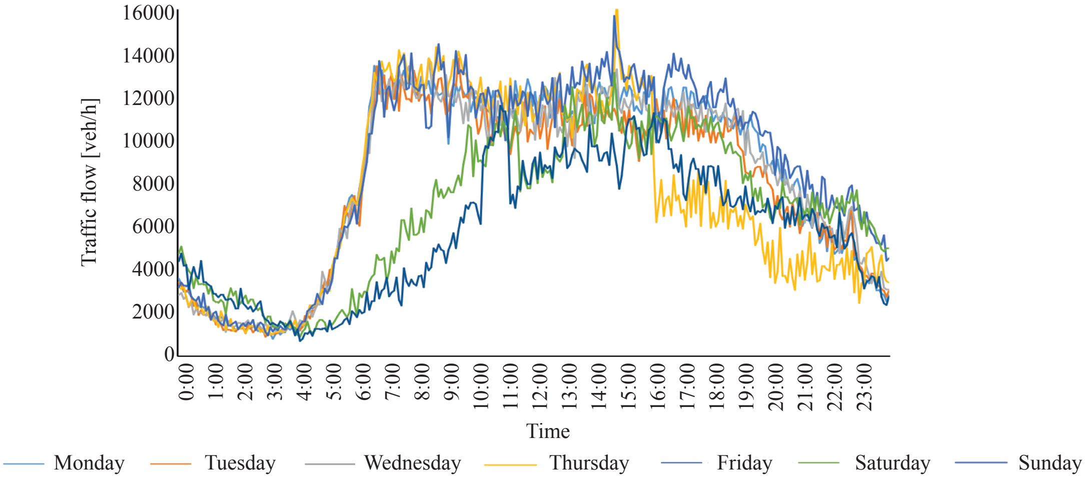


Figure 3 – One week's traffic flow at a VDS

urban population. The trend for non-working days was slightly different. Even though the variation trend of the working days and the non-working days was slightly different, the research [14–16] showed that even when the two were not distinguished, the prediction model performance under the deep learning model did not have a significant loss as compared to the complete distinction. Because of the contextual relationship of the data, the traffic flow at any moment could be a continuous time series without distinguishing the two; otherwise, the correlation between the two would be severed. Therefore, in the subsequent experiments, both were used as the input.

In general, the quality and validity of the original data are directly affected by the accuracy of the acquisition equipment, the reliability of the transmission media, and security of the storage equipment. The traffic flow data collected directly inevitably have abnormal data. It is necessary to pre-process the dirty data. For threshold determination of traffic flow detection data, a reasonable flow range is usually defined as *Equation 1*:

$$0 \le v \le f_c \cdot C \cdot \frac{T}{60} \tag{1}$$

where v is the flow, C is the upper limit of road capacity (v/h), T is the interval time (min), and f_c is the correction coefficient, which is generally between 1.3 and 1.5 [17].

3. RESEARCH METHOD

3.1 Technical route

Figure 4 shows the main technical flow of this study. The spatial distribution features of various VDSs and the bidirectional context-dependent time features of the traffic flow data were fully considered. After applying the data cleaning rules like *Equation 1* and data recovery, a reliable data set is obtained. Besides, it is necessary to do data analysis

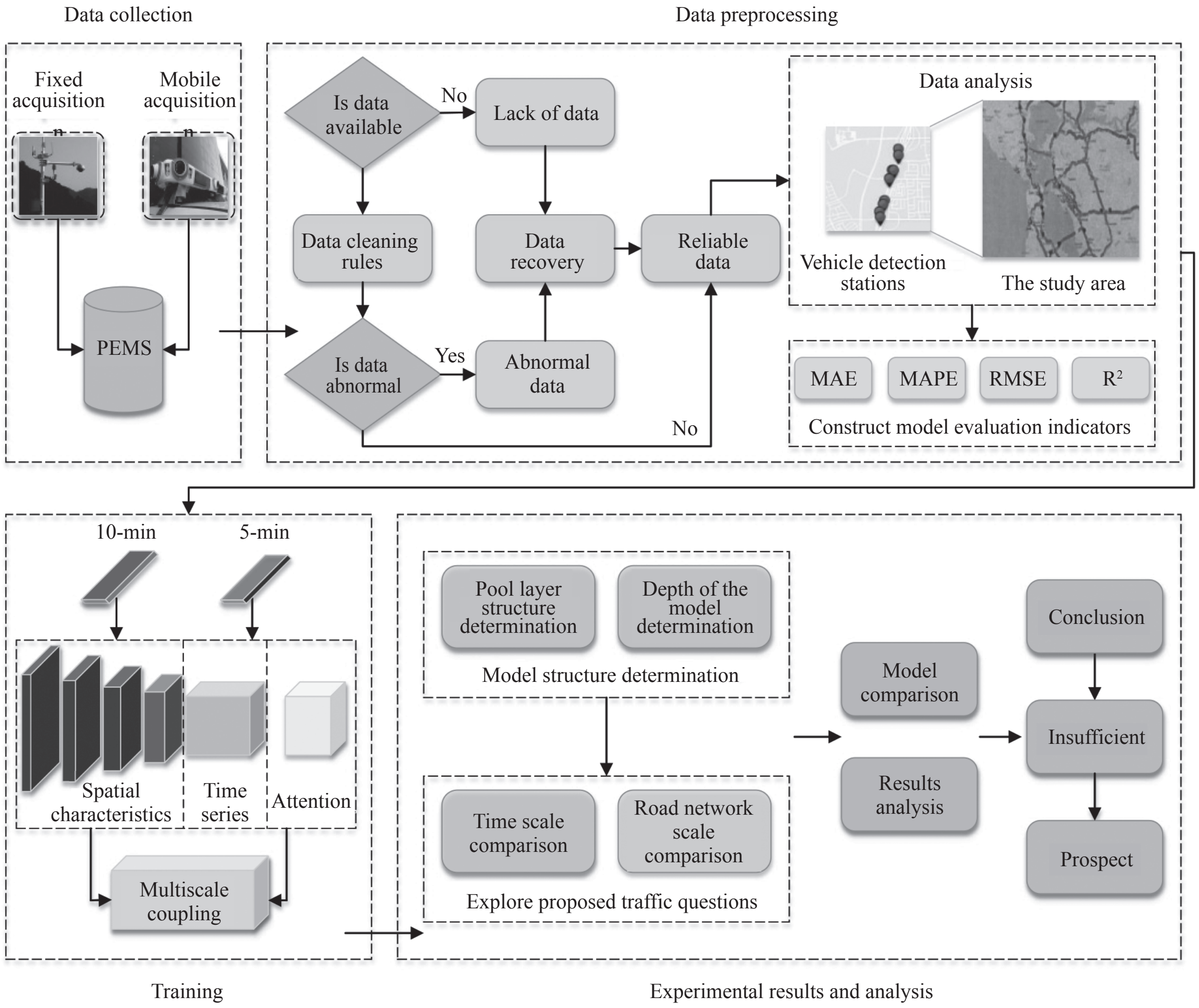


Figure 4 – Research flowchart

to make sure the data are qualified to train the deep learning model. The experiment is mainly divided into three parts: exploring the optimal model structures, exploring whether the proposed model solves the proposed problem, and horizontally comparing the prediction accuracy with other models.

3.2 Proposed model

Inspired by the layered structure of neurons, the proposed BCM-Net is designed to enhance the performance of spatiotemporal features extraction, which includes three main parts. First, the spatial features of short-term traffic flow are extracted by the 1D convolution neural network. The BCA block is composed of Bidirectional Gated Recurrent Unit (BiGRU) and feed-forward self-attention mechanism. By incorporating the BCA block, the feature extraction network is enriched in the bidirectional structure and the high-dimensional features are integrated. Through the modification of the loss function, the model is further optimised. Finally, the IBP sub-network is used as a multi-scale merging method, which is aimed at integrating multi-scale information and improving the robustness of the prediction model. *Figure 5* shows the architecture of the

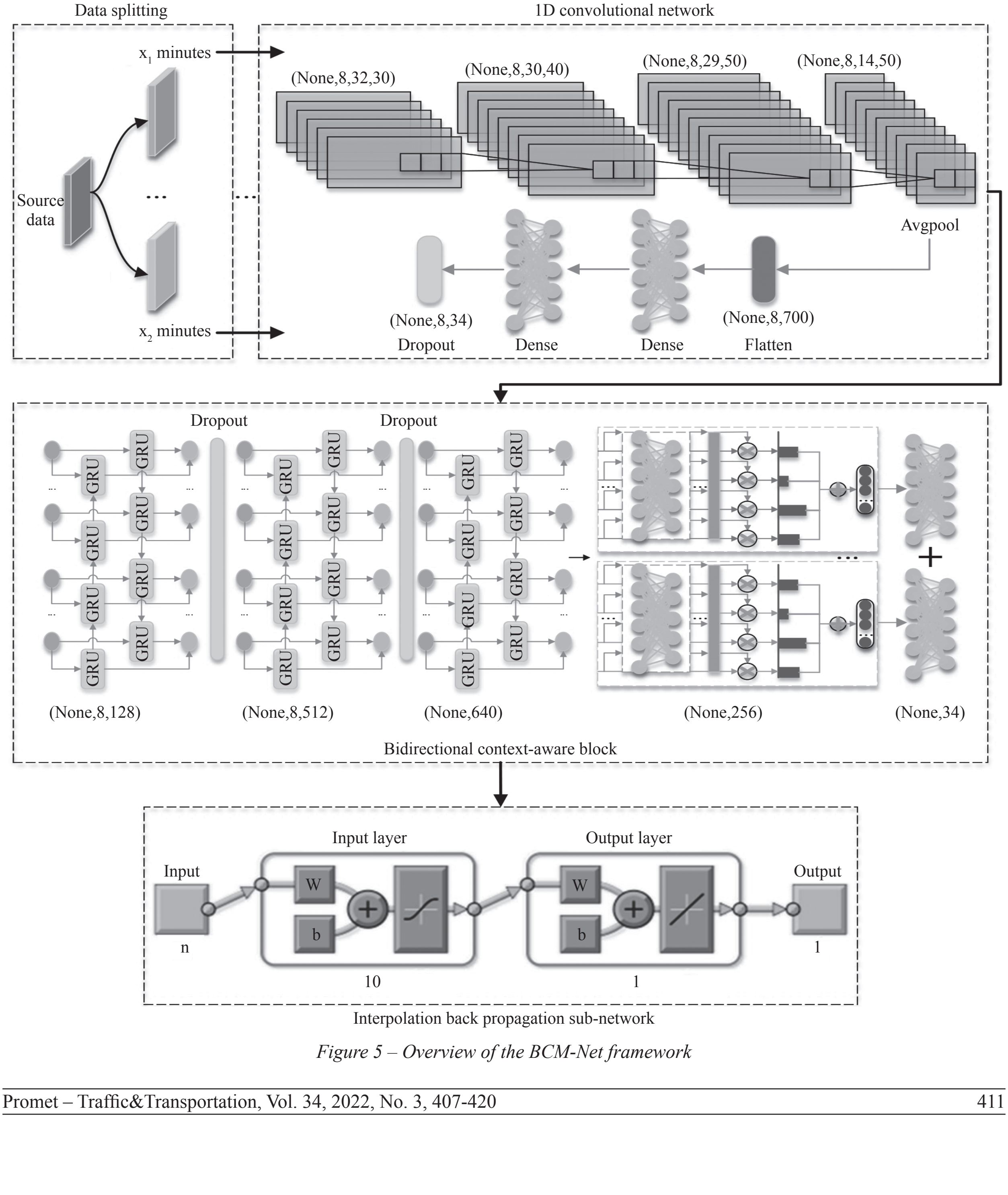


Figure 5 – Overview of the BCM-Net framework

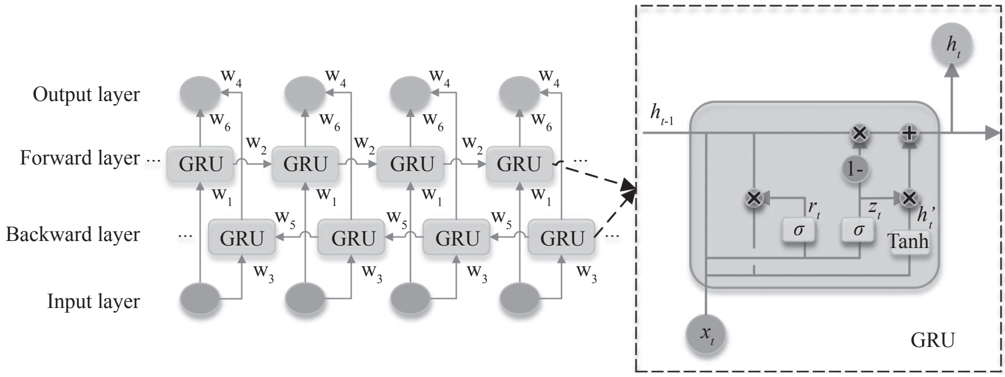


Figure 6 – Bidirectional gated recurrent uni

proposed framework. In this research, a grid search method is used to determine the hyper-parameters. The basic idea is to divide the interval of the optimisation parameter with a grid. We calculated and evaluated the prediction performance of the model on each grid node in turn, and determined whether the hyper-parameters are optimal by comparing the quality of the model. The remaining hidden layer parameters are determined by the Root Mean Square Prop (RMSProp) gradient optimisation algorithm [18].

The spatial characteristics of the traffic flow data have more of time-series correlation caused by spatial correlation, rather than the traditional spatial correlation among various VDSs. In order to extract the spatial features, a 1D convolutional neural network was built. The 1D convolutional neural network comprises three 1D convolutional layers and a 1D pool layer. We use 8*34*1 vectors for each time scale as the input of the network. The first parameter represents the length of the time series. The parameters setting of the three 1D convolution layers are 3*3*30, 3*3*40, and 2*2*50, respectively. Through the flatten layer and the fully connected layer network, the (None,8,34) tensor was output to the next part, where None meant the batch size. Meanwhile, a parameter regularisation layer was added to avoid over-fitting with its value set to 0.2 [18]. We abandoned the traditional Rectified Linear Unit (ReLU) activation function, and adopted a Scaled-Expected Linear Units (SELU) activation function to provide the convolutional network with the self-normalisation function for better convergence and more effectiveness in avoiding the disappearance of the gradient [19].

Bidirectional Gated Recurrent Unit

Because the traffic flow data is not only affected by the past but also by the future, the (None,8,34) tensor was put into the BiGRU. The LSTM or Gated Recurrent Unit (GRU) is usually used to capture the temporal series features, which ignores the bidirectional context relevance of traffic flow. In this study, a BiGRU network was constructed, which was superimposed by three BiGRU layers with nodes of 64, 256, and 320 respectively. Over-fitting was avoided by adding the parameter regularisation layer between layers with the value of 0.2. Finally, the sub-network output the (None, 640) tensor to the next part. The LSTM model realizes the functions of memorising, forgetting, and updating information through the structure of gates. Because the gate structures are introduced, the training time of the LSTM is much longer than that of the RNN, which is not beneficial to the rapid modelling of large road network. As a compromise, the GRU model is proposed. As shown in *Figure 6*, the GRU model, as a variant of the LSTM, has two differences from LSTM. Firstly, the GRU model simplifies the two information flows of the LSTM model into one. Secondly, the GRU model combines the forgetting gate and the input gate into a single update gate and introduces a reset gate [20].

Now, consider that r_t represents the value of the reset gate and z_t represents the value of the update gate. Then, the reset gate and the update gate inside the gated recurrent unit can be represented by *Equations 2 and 3*:

$$r_t = \sigma(w_{ir}x_t + b_{ir} + w_{hr}h_{t-1} + b_{hr}) \tag{2}$$

$$z_t = \sigma(w_{iz}x_t + b_{iz} + w_{hz}h_{t-1} + b_{hz}) \tag{3}$$

where x_t is the input value, and h_{t-1} is the hidden state of the cell at the previous moment, w_{ir} and w_{iz} are the input weights of the two gates, w_{hr} and w_{hz} are the weight of the hidden layers of the two gates, b_{ir} and b_{iz} are the input bias terms of the two gates, b_{hr} and b_{hz} are the hidden layer bias terms of the two gates, and σ is the sigmoid activation function.

The new cell states and the candidate cell states of the GRU are represented by h_t and h'_t, as shown in *Equations 4 and 5*:

$$h_t = z_t \cdot h'_t + (1 - z_t) \cdot h_{t-1} \tag{4}$$

$$h'_t = \tanh(w_{in}x_t + b_{in} + r_t \cdot (w_{hn}h_{t-1} + b_{hn})) \tag{5}$$

where w_{in} is the input weight of the candidate cell, w_{hn} is the hidden layer weight of the candidate cell, and "*" is the Hadamar product.

In the traditional RNN, the model represents features association by propagating the output of the former cell to the latter cell. However, we often only consider the one-way features and ignore the impact of subsequent data on previous data. We can effectively mine the bidirectional features of time series data by means of two-way propagation. The BiGRU model has been widely used in the field of natural language processing and has been proven to be effective in the predictive analysis of the contextual text [21, 22].

The BiGRU has two propagation layers: forward layer and backward layer. The forward propagation layer carries out the forward calculation from the initial moment to the moment t to obtain the output h_t of the forward hiding layer at each moment. The back propagation layer performs the reverse calculation from time t to the initial time to obtain the output h'_t of the backward hiding layer at each moment. Combined with the output of the forward propagation layer and the back propagation layer at each moment, the final output y_t is calculated, where x_t represents the input at time t, w_1 to w_6 represents the shared weight values, and the calculation process is shown with *Equations 6–8*.

$$h_t = f(w_1x_t + w_2h_{t-1}) \tag{6}$$

$$h'_t = f(w_3x_t + w_5h'_{t+1}) \tag{7}$$

$$y_t = f(w_4h_t + w_6h'_t) \tag{8}$$

Feed-forward self-attention mechanism

By building the feed-forward self-attention mechanism, the internal connections of high-dimensional feature were integrated. The self-attention layer was constructed by the softmax classifier and the fully connected neural network. Through the BiGRU, the higher-dimensional eigenvectors were obtained. In order to better characterise the relationship between these eigenvectors, we need a function to calculate the weight of these eigenvectors. According to the basic knowledge of probability theory, the function must satisfy two conditions:

1) All values are greater than 0,
2) The sum of all values is equal to 1.

Therefore, the softmax classifier was used to optimise the weight ratio of each internal time feature vector. In addition, through the fully connected network part, we could also complete feature dimensionality reduction. As shown in *Figure 7*, according to the time series features h_1-h_i obtained by the BiGRU, the model mines the internal correlation of the feature vectors and outputs the periodic time series features in a weighted average manner to improve the accuracy of the traffic flow prediction model. The main task of the self-attention mechanism is to integrate i input eigenvectors into a new i-dimensional eigenvector H. The algorithm is divided into three steps:

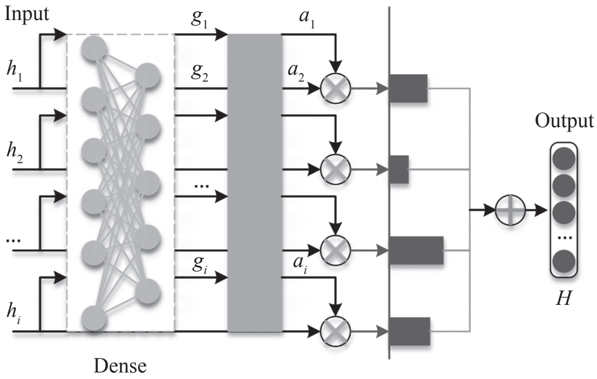


Figure 7 – Feed-forward self-attention mechanism

1) Input i feature vectors h_1-h_i into a fully connected neural network for training, get the evaluation function G, evaluate the influence of each feature vector on H, and then get the evaluation value gi corresponding to each feature vector.
2) Normalise the corresponding weight value a_i of each feature vector through by using the softmax classifier.
3) Use the weighted average method to obtain H. The calculation process is shown with *Equations 9–11*.

$$g_i = G(h_i) = \tanh(w^T h_i + b_i) \tag{9}$$

$$a_i = \text{Softmax}(g_i) = \frac{e^{g_i}}{\sum_j e^{g_j}} \tag{10}$$

$$H = \sum_{i=1}^{n} a_i \cdot h_i \tag{11}$$

Interpolation back propagation sub-network

In order to merge the multi-scale time features of the traffic flow data, the IBP sub-network was designed. The input was the *n*-dimensional vector consisting of *n* different time scales traffic flow predicted values with the ratio of the training set, the test set, and the verification set was 0.7:0.15:0.15, the hidden layer neuron was set to 10, and the training optimisation algorithm adopted the Levenberg-Marquardt (LM) algorithm [23]. In previous short-term traffic flow prediction models, the traffic flow data of a single time scale was usually the only input data. However, the traffic flow data collected in reality can divide into different levels. Data sources with a single time scale find it difficult to represent the unique time series characteristics of each scale, thus limiting the forecast vertex of the model. The existing literature [24, 25] shows that the large-scale data can help the model in predicting the overall trend of traffic flow, and the small-scale of traffic flow data can help to capture the small fluctuations. However, the very intensive time scale tends to contain a considerable amount of Gaussian noise, making it difficult for the prediction model to capture the time features.

Suppose two time scales are taken as examples with the small scale is x_1-min, and the large scale is x_2-min. The goal is to merge these two scales to the *z*-min scale. We used the unit time that is 1-min as the standard for interpolation. After selecting the items that meet the target scale and combining them into 2-dimensional vectors, we input them into the BP neural network. Further generalise the situation to *n* time scales, as shown in *Equation 12*:

$$z_i = f(x_1^i, x_2^i, \ldots, x_n^i), \quad n = 1,2,\ldots \tag{12}$$

where *z* represents the target scale, *i* represents the moment *i*, and *f* represents the mapping relationship.

The outstanding advantages of the IBP sub-network are its strong nonlinear mapping ability and flexible network structure. The IBP sub-network forward transfer process is as follows: the weight between node *i* and node *j* is w_{ij}, the threshold value of node *j* is b_j, and the output value of each node is x_j. The output value of each node is realised according to the output value of all nodes in the upper layer, the weight value of the current node and all nodes in the upper layer, the threshold value of the current node and the activation function. The specific calculation formula is shown with *Equations 13 and 14*, where *m* is the number of nodes in the input layer.

$$S_j = \sum_{i=0}^{m-1} w_{ij} x_i + b_j \tag{13}$$

$$x_j = f(S_j) \tag{14}$$

Loss function optimisation

The model adopts the sum of the MSE and the regularisation term as the loss function, that is, adding the structural risk representing the complexity of the model to the empirical risk. The loss function can be expressed as shown in *Equations 15 and 16*:

$$L = \frac{1}{N}\sum_{i=1}^{N} |x_i - y_i|^2 + \frac{1}{2}\lambda \| W_i \|^2 \tag{15}$$

$$\lambda = \prod_i weight\_decay_i \tag{16}$$

where *N* is the number of samples, *x* represents the measured value at the moment *i*, and *y* represents the predicted value at the moment *i*. Among them, the first term is the MSE loss function, the second term is the L2 regular term of the weight, and *λ* is the regular term coefficient, determined by the product of the attenuation coefficients of each weight. By introducing the regular term as a penalty term of the loss function, the model balances the empirical risk and the complexity of the model, which can effectively prevent the phenomenon of over-fitting.

4. RESULTS AND ANALYSIS

The data of 34 VDSs in the PEMS database were selected for training, the optimised MSE was the loss function, batch size was 64, and epoch was 300. In this part, the different pool layer structures and different depth of the models were trained to determine the optimal model structure. When we determined the best model structure, an experiment was designed to explore the impact of different time scales and different road network sizes on prediction accuracy. Moreover, the chosen classic prediction models were compared with the proposed model.

4.1 Evaluation metrics

Traffic flow prediction models usually need to meet the requirements of accuracy, real-time performance, and robustness. In order to judge whether the traffic flow prediction model is qualified, the average absolute error (MAE), mean absolute percentage error (MAPE), root mean square error (RMSE), and goodness of fit (R^2) were used to evaluate the models. The smaller the values of MAE, MAPE, and RMSE were, the higher was the accuracy. The closer R^2 was to 1, the better the fitting effect was. The calculation process is shown with *Equations 17–20*.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|x_i - y_i| \tag{17}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\frac{|x_i - y_i|}{x_i} \tag{18}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}|x_i - y_i|^2} \tag{19}$$

$$R^2 = 1 - \frac{\left(\sum_{i=1}^{N}|x_i - y_i|\right)^2}{\sum_{t=1}^{N}|x_i - y_i|^2} \tag{20}$$

where, x_i represents the real traffic flow data at moment i, y_i represents the predicted traffic flow data output by the model at moment i, and N represents the length interval of the time series.

4.2 Model structure determination

As shown in *Table 1*, in order to avoid the over-fitting problem, the 1D maximum pooling layer was built with 2 kernel size and 1 stride, between the second and the third 1D convolutional layers. The 1D average pooling layer was also built, whereby kernel size was 2 and stride was 1, after the third 1D convolution layer. It turned out that the prediction model using the average pooling method had a higher prediction accuracy. The 1D maximum pooling layer was suitable for reducing useless and redundant information, while the 1D average pooling layer was suitable for preserving and synthesising the overall data characteristics, which was more needed by our model.

Table 1 – Pool layer structure comparison (the best results are highlighted

Model structure		Metrics			
		MAE	MAPE	RMSE	R^2
BCM-Net	Maxpool	21.50	13.86%	30.72	97.84%
	Avgpool	18.36	11.54%	26.78	98.21%

Table 2 – Depth of the model comparison (the best results are highlighted)

Model structure		Index			
Conv1D	BiGRU	MAE	MAPE	RMSE	R^2
2	1	22.36	14.67%	32.73	95.13%
	2	22.24	14.34%	32.46	95.74%
	3	21.19	13.11%	31.59	96.02%
	4	23.76	13.96%	31.92	95.86%
3	1	21.89	13.65%	32.47	96.81%
	2	20.83	13.52%	29.94	97.64%
	3	18.36	11.54%	26.78	98.21%
	4	19.64	12.47%	27.15	97.56%

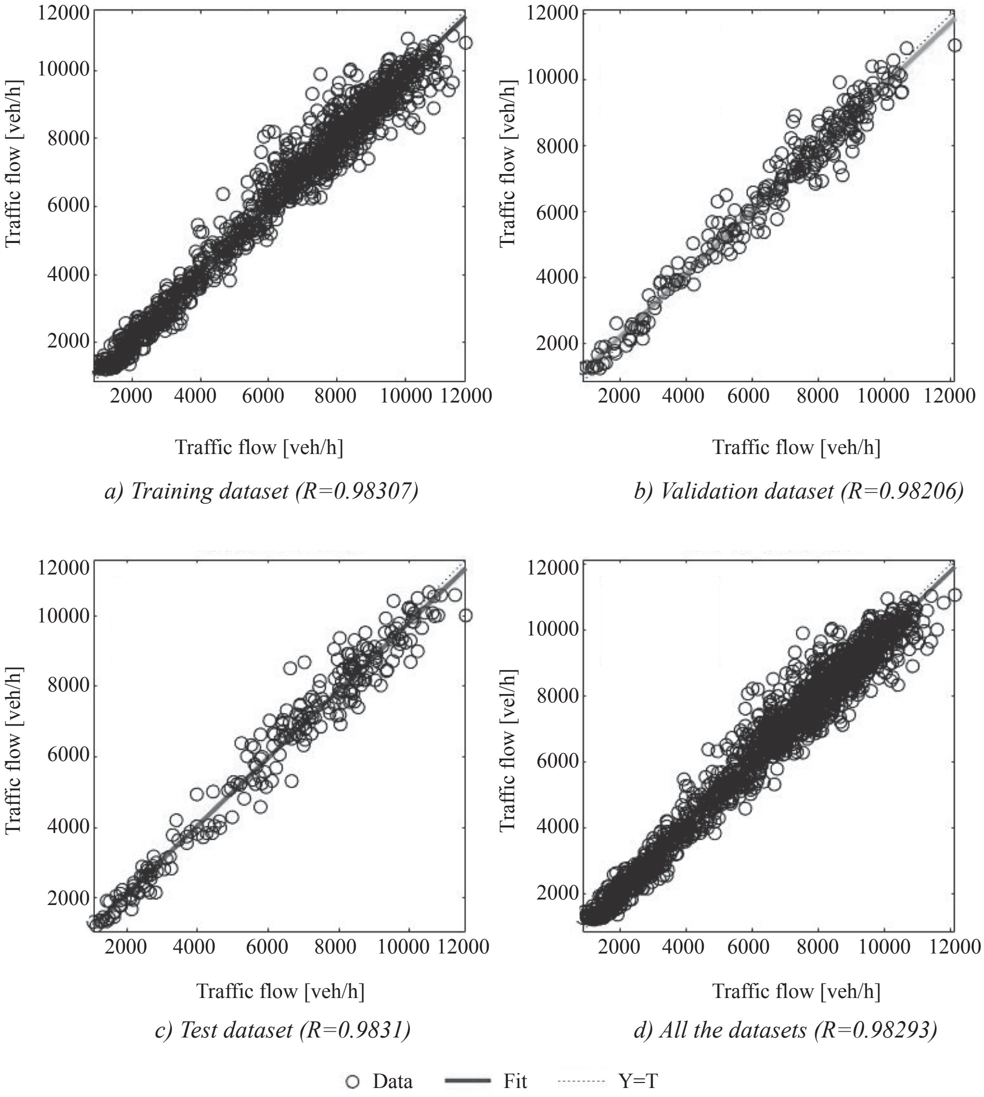
Since different volumes and different dimensions of training sets mean different complexities of prediction problems, in this study, we explored the influence of different layers of BiGRU on the prediction accuracy for different numbers of convolution layers through controlled experiments. *Table 2* showed that the prediction accuracy was the highest when the number of 1D convolution layers was 3 and the number of BiGRU layers was 3. Therefore, the BCM-Net model designed in this study adopted the above three-layer superposition structure.

4.3 Time scale comparison

As mentioned in section 3.2.3, small-scale data would be affected by noise and large-scale data would lose detailed information. Thus, we need to figure out which time scale data were qualified for multi-scale merging. The minimum time scale that the PeMS database could provide was 5 minutes, from which we derived 10-min and 15-min data intervals. As show in *Table 3*, in the BCM-Net model without multi-scale merging, the result showed that

Table 3 – Time scale comparison (the best results are highlighted)

Model		Index			
		MAE	MAPE	RMSE	R^2
BCM-Net (without multi-scale merging)	5-min	20.39	12.61%	28.94	97.24%
	10-min	21.16	12.82%	29.37	97.79%
	15-min	27.63	14.97%	34.86	95.16%



a) Training dataset (R=0.98307)

b) Validation dataset (R=0.98206)

c) Test dataset (R=0.9831)

d) All the datasets (R=0.98293)

Figure 8 – Scatter plots of multi-scale merging

the 5-min and 10-min data still held the prediction accuracy pretty well, but the 15-min data prediction accuracy was far inferior to the previous two. That was because as the time scale increases, too much detailed information was discarded and the data set had also become smaller due to the dilution of the data. Therefore, we chose 5-min and 10-min data to merge. In order to compare with the true value, the target scale was 5-min.

In the scatter plot, the 45° inclination of the regression line indicated that the true value was equal to the predicted value. As shown in *Figure 8*, the overall R^2 of training set, test set, and verification set all reached above 0.98. The overall prediction R^2 of the multi-scale prediction model was 98.29%, which was 1.05% higher than that of the 5-min scale BCM-Net model. This showed that the IBP sub-network could effectively merge the multi-scale information of time series and improve the prediction accuracy.

4.4 Road network scale comparison

In order to carry out this experiment, we added data of 6 VDSs on the basis of the previous 34 VDSs. The 40 VDSs were divided into different datasets with different road network scales. As show in *Figure 9*, our model did not lose too much accuracy in different road network scales. It was foreseeable that the model performed best in the 34 VDSs scale. The small-scale road network did not require too deep models to train. The high-dimensional feature vector generated by such a model could not better characterise the road network data at a small scale. Obviously, we could also adjust the

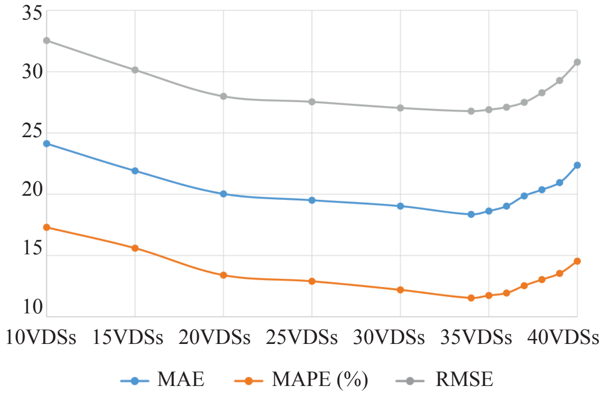


Figure 9 – Road network scale comparison

depth of our model to adapt to small road networks. This did not mean that blindly increasing the depth could solve the problems caused by the increase in the scale of the road network. As the scale of the road network increases, the dimensionality of the input data was also increasing. The model would fall into the curse of dimensionality. Ideally, a large enough data set can make the model training with good enough accuracy, but the training process will become extremely difficult. As we can see in the *Figure 10*, our model's training accuracy gradually decreased after 34 VDSs. Compared with the current prediction models of about 10 VDSs, the model under the scale of about 30 VDSs was sufficient to meet the needs of ordinary road networks.

In a large road network, the training efficiency of the model was also an issue worthy of attention. As shown in *Table 4*, the BCM-Net (LSTM) was constructed with the three-layer bidirectional Long-Short-Term Memory network, whose nodes are 64, 256, and 320 respectively. As the GRU was the simplified version of LSTM, the BCM-Net (GRU) reduces 12 544, 197 120, and 533 120 parameters in each layer respectively, compared with the BCM-Net (LSTM) model. Accordingly, the BCM-Net (GRU) model could achieve convergence with less epoch quantity and shorter training time. Therefore, the BCM-Net (GRU) model sacrificed a little accuracy in exchange for almost 1/3 of improvement in training efficiency, which was inevitable. In large traffic road network, the model had the advantages of fewer parameters, easier fitting, less sample requirements, and less training time.

4.5 Model comparison

We compared the BCM-Net model proposed with the classical GRU, LSTM, auto-encoders (SAEs), BiGRU, and Convolution Bidirectional Gated Recurrent Unit (Conv-BiGRU) models. Among them, GRU, LSTM, BiGRU, and Conv-BiGRU models had the same number of nodes as the corresponding parts of the model proposed in this paper. The SAEs model set five auto-encoder layers and a fully connected layer with 34 nodes. As shown in *Table 5*, the indicators of the BCM-Net proposed in this paper were all superior to the traditional prediction methods, with the higher prediction accuracy of R^2, which was 98.21%, and the smaller error indexes of MAE, MAPE, and RMSE, which were 18.36, 11.54% and 26.78 respectively.

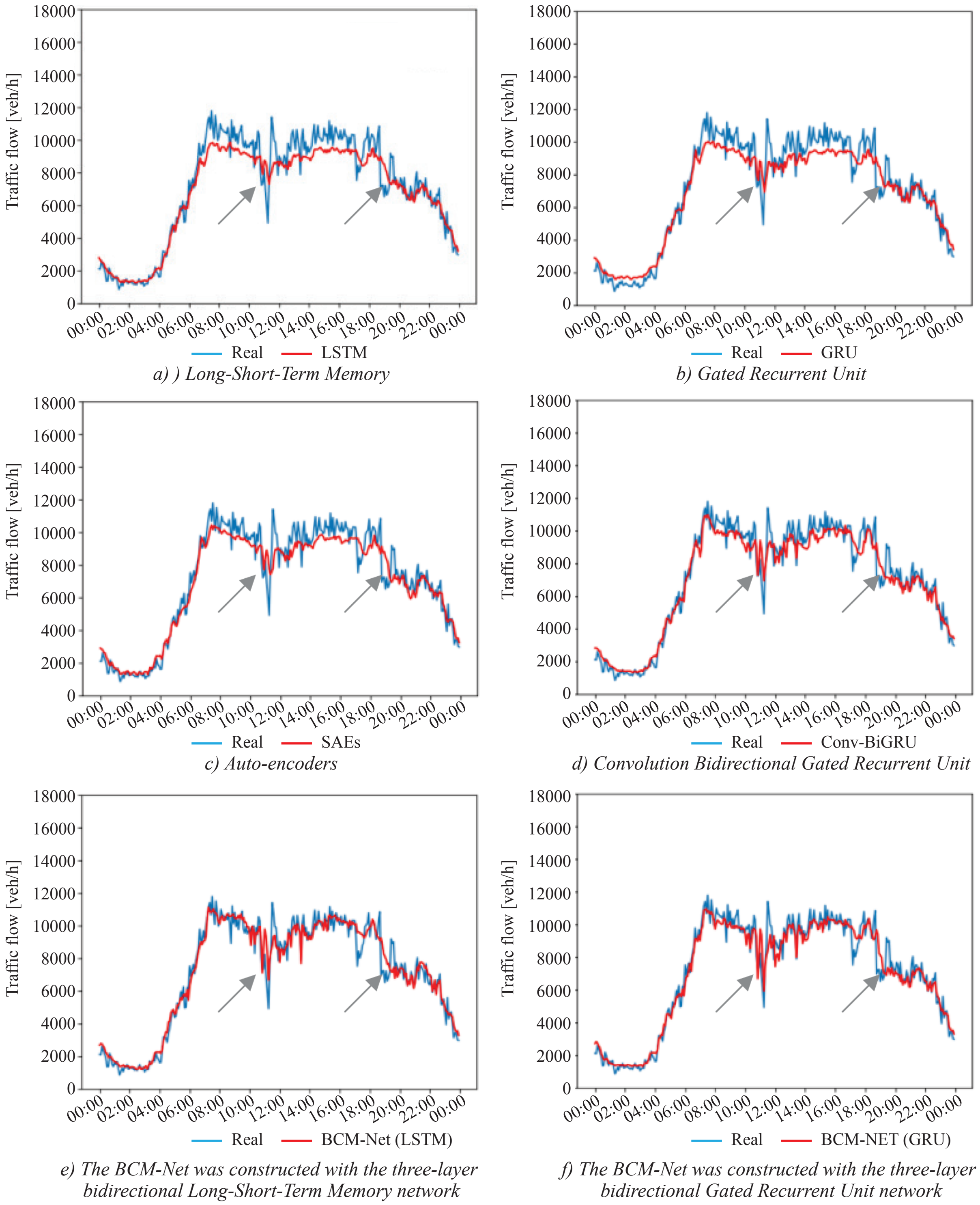
We could see that the GRU model performed worse than the Conv-GRU model because it was lacking the expression of the spatial characteristics. As we put BiGRU into the Conv-GRU model, the Conv-BiGRU model performed better in various indicators. Meanwhile, feed-forward self-attention mechanism also played the same role in improving accuracy. Although the LSTM and GRU models completed the functions of “forgetting” and “updating” through the gate structure and had a good representation of the time series time characteristics, they still lack the representation of the context

Table 5 – Model comparison (the best results are highlighted)

Model	Index			
	MAE	MAPE	RMSE	R^2
LSTM	27.37	15.57%	39.14	94.76%
GRU	27.96	15.78%	39.85	94.85%
SAEs	27.98	16.53%	39.36	94.89%
Conv-GRU	24.61	14.43%	35.83	95.63%
Conv-BiGRU	22.13	13.62%	31.96	96.67%
BCM-Net	18.36	11.54%	26.78	98.21%

Table 4 – BCM-Net (GRU) compare with BCM-Net (LSTM) – the best results are highlighted

Model	Parameters of the three-layer recurrent neural network	Saturation epoch	Training time	Index			
				MAE	MAPE	RMSE	R^2
BCM-Net (LSTM)	50 176, 788 480, 2 132 480	267	6h23min	17.41	11.32%	26.35	98.43%
BCM-Net (GRU)	37 632, 591 360, 1 599 360	223	4h45min	18.36	11.54%	26.78	98.21%



a)) Long-Short-Term Memory

b) Gated Recurrent Unit

c) Auto-encoders

d) Convolution Bidirectional Gated Recurrent Unit

e) The BCM-Net was constructed with the three-layer bidirectional Long-Short-Term Memory network

f) The BCM-Net was constructed with the three-layer bidirectional Gated Recurrent Unit network

Figure 10 – Partial fitting curve contrast

relevance and periodicity of the time series data. The BCA block, which is composed of BiGRU and feed-forward self-attention mechanism, was designed to make up for the deficiency. Therefore, the BCM-Net model proposed in this paper could more fully capture the temporal and spatial characteristics of the traffic flow data hidden.

The fitting curve of the experimental part was shown in *Figure 10*. An interesting thing happened in the test set. We could see that the predict curve of the BCM-Net (GRU) and the BCM-Net (LSTM) model almost matched the curve of the true value, when the true value changed significantly from 10 a.m. to 12 a.m. That meant that the model could predict major changes in traffic flow to a certain extent, while the traditional model performed poorly. But the results showed that none of the models predicted a sudden drop in real numbers from 6 p.m. to 8 p.m. We had reason to speculate that emergencies occurred at this time that caused an abnormal change in traffic flow. Therefore, we could conclude that our model could predict major changes in regular traffic flow, such as a short-term surge in traffic flow caused by major events, or a surge in traffic flow during rush hours. But if an accident occurs at a certain moment that causes a sudden change in traffic flow, this is beyond the scope of the model's predictive ability, which is data-driven. We could put manual intervention in this situation to constrain our model and ensure that such situations are included in the training data set.

5. CONCLUSIONS

This study proposed a BCM-Net to predict the short-term traffic flow data. In the BCM-Net, the feature extraction part combined BCA blocks to enhance the high-dimensional spatiotemporal feature, the IBP sub-network was applied to extract multi-scale traffic flow features, and by using optimised loss function to make the network find the optimal value in the weight space. The proposed method was first evaluated on a created time-scale and spatial-scale real world datasets. The results showed the advantages of the proposed method in processing multi-scale data in large road network scale. To drive the state-of-the-art models, all models are compared in the same data set in the same research area. The experimental results confirmed that the proposed framework performs better in handing complex traffic flow data with other state-of-the-art approaches. It shows that it can predict multi-scale short-term traffic flow data timely and accurately on a large road network scale, which can be used for the overall management of the ITS and emergency forecasts.

However, the model structure still needs to be further optimised. In large traffic networks, the transportation network will be more complicated. The way of simply stacking each VDS to form a multi-dimensional input still has limitations, which is suitable for rapid modelling and generalisation. In this way, we ignore the topology of the road network. In a specific area, if we want to improve the prediction accuracy of the model, we need to try to combine with GCN to optimise the representation of spatial features. In addition, we will combine the multi-scale with the attention mechanism to give weight to diverse time-scale data.

陈桂兴[1]，郑贵洲[1]（通信作者）
[1] 中国地质大学（武汉）地理与信息工程学院，
空间规划与人地系统模拟研究中心，武汉，430074

基于双向上下文感知和多尺度融合的短期交通流量预测混合网络

摘要

短期交通流预测是指在提取路网时空特征的基础上，自动预测未来一段时间内的交通流变化。对政府而言，实时准确的预测交通流量对于规划道路管理、提高交通效率至关重要。近年来，深度学习技术在短期交通流量预测领域取得了显著进展。然而，以往基于深度学习的预测方法主要局限于时间特征，迄今为止未能有效挖掘数据内部的双向上下文时空特征。此外，这些模型的精度和实用性受到路网规模和单一时间尺度的局限性。针对这些问题，本文提出了一种基于双向上下文感知的多尺度融合的混合网络。该网络提出一种能够实时准确预测交通流变化的新型短期交通流预测框架。模型在特征提取结构中加入了双向上下文感知模块，有效地整合了时空特征；采用反向传播插值子网络耦合多尺度信息，进一步提高模型的鲁棒性。在不同数据集上的实验结果表明，该方法的性能优于现有方法。

关键词
智能交通系统；门控循环单元；短期交通流量预测；双向上下文感知；插值-反向传播算法

REFERENCES

[1] Agachai S, Wai HH. Smarter and more connected: Future intelligent transportation system. *IATSS Research*. 2018;42: 67-71. doi: 10.1016/j.iatssr.2018.05.005.

[2] Liu Z, et al. Effect of time intervals on K-nearest neighbors model for short-term traffic flow prediction. *Promet – Traffic&Transportation*. 2019;31(2): 129-139. doi: 10.7307/ptt.v31i2.2811.

[3] Liu Z, et al. A hybrid short-term traffic flow forecasting method based on neural networks combined with K-nearest neighbor. *Promet – Traffic&Transportation*. 2018;30(4): 445-456. doi: 10.7307/ptt.v30i4.2651.

[4] Hubel DH, Wiesel TN. Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology.* 1959;148(3): 574. doi: 10.1113/jphysiol.1959.sp006308.

[5] Mou L, Zhao P, Xie H, Chen Y. T-LSTM: A long short-term memory neural network enhanced by temporal information for traffic flow prediction. *IEEE Access*. 2019;7: 98053-98060. doi: 10.1109/ACCESS.2019.2929692.

[6] Doan E. Short-term traffic flow prediction using artificial intelligence with periodic clustering and elected set. *Promet – Traffic & Transportation*. 2020;32(1): 65-78. doi: 10.7307/ptt.v32i1.3154.

[7] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Computation*. 1997;9(8): 1735-1780. doi: 10.1162/neco.1997.9.8.1735.

[8] Wei W, Wu H, Ma H. An AutoEncoder and LSTM-based traffic flow prediction method. *Sensors*. 2019;19(13): 2946. doi: 10.3390/s19132946.

[9] Zheng H, Lin F, Feng X, Chen Y. A hybrid deep learning model with attention-based conv-LSTM networks for short-term traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(11): 6910-6920. doi: 10.1109/TITS.2020.2997352.

[10] Qiao Y, Wang Y, Ma C, Yang J. Short-term traffic flow prediction based on 1DCNN-LSTM neural network structure. *Modern Physics Letters B*. 2020;35(2): 2150042. doi: 10.1142/S0217984921500421.

[11] Li Z, et al. A hybrid deep learning approach with GCN and LSTM for traffic flow prediction. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2019. doi: 10.1109/ITSC.2019.8916778.

[12] Zhang Y, Yang SM, Xin DR. Short-term traffic flow forecast based on improved wavelet packet and long short-term memory combination model. *Transportation Systems Engineering and Information*. 2020;20(2): 208-214. doi: 10.1109/CSAE.2011.5953161.

[13] Shan G, Ye Z, Guo QC. Study on exhaust emission test of diesel vehicles based on PEMS. *Procedia Computer Science*. 2020;166: 428-433. doi: 10.1016/j.procs.2020.02.070.

[14] Zheng L, et al. Dynamic spatial-temporal feature optimization with ERI big data for short-term traffic flow prediction. *Neural Computation*. 2020;412: 339-350. doi: 10.1016/j.neucom.2020.05.038.

[15] Doan E. LSTM training set analysis and clustering model development for short-term traffic flow prediction. *Neural Computing and Applications*. 2021;4: 1-14. doi: 10.1007/s00521-020-05564-5.

[16] Kang DQ, Lv YS, Chen YY. Short-term traffic flow prediction with LSTM recurrent neural network. *IEEE International Conference on Intelligent Transportation Systems (ITSC)*. 2018. doi:10.1109/ITSC.2017.8317872.

[17] Gal Y, Ghahramani Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. *JMLR.org*. 2015.

[18] Zhang H, et al. Power control based on deep reinforcement learning for spectrum sharing. *IEEE Transactions on Wireless Communications*. 2020;19(6): 4209-4219. doi: 10.1109/TWC.2020.2981320.

[19] Pan X, et al. Identifying patients with atrioventricular septal defect in down syndrome populations by using self-normalizing neural networks and feature selection. *Genes*. 2018;9(4): 208. doi: 10.3390/genes9040208.

[20] Fu R. Using LSTM and GRU neural network methods for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*. 2016. doi:10.1109/YAC.2016.7804912.

[21] Oh YR, Park K, Jeon HB, Park JG. Automatic proficiency assessment of Korean speech read aloud by non-natives using bidirectional LSTM-based speech recognition. *ETRI Journal*. 2020;42(10). doi: 10.4218/etrij.2019-0400.

[22] Zeyer A, et al. A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition. *IEEE ICASSP*. 2017. doi: 10.1109/ICASSP.2017.7952599.

[23] Hu G, Feng ZZ, Cao J, Huang H. Nonlinear calibration optimization based on the Levenberg-Marquardt algorithm. *IET Image Processing*. 2020;14(7). doi: 10.1049/iet-ipr.2019.1489.

[24] [24] Dai X, et al. Deeptrend 2.0: A light-weighted multi-scale traffic prediction model using detrending. *Transportation Research Part C Emerging Technologies*. 2019;103(1): 142-157. doi: 10.1016/j.trc.2019.03.022.

[25] Huang H, et al. Effect of multi-scale decomposition on performance of neural networks in short-term traffic flow prediction. *IEEE Access*. 2021;9: 50994-51004. doi: 10.1109/ACCESS.2021.3068652.