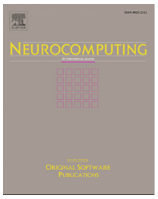
Neurocomputing 427 (2021) 50–63

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Traffic flow prediction over muti-sensor data correlation with graph convolution network

Wei Li [a], Xin Wang [a], Yiwen Zhang [a,*], Qilin Wu [b,c]

[a] *School of Computer Science and Technology, Anhui University, Hefei 230000, China*
[b] *School of Information Engineering, Chaohu University, Chaohu 238000, China*
[c] *School of Management and Engineering, Nanjing University, Nanjing 210093, China*

ARTICLE INFO

Article history:
Received 15 March 2020
Revised 17 June 2020
Accepted 10 November 2020
Available online 1 December 2020
Communicated by Zidong Wang

Keywords:
Multisensor data correlation
Graph convolutional network
Spatial–temporal correlation
Traffic flows prediction

ABSTRACT

Accurate and real-time traffic flow prediction plays an important role in improving the traffic planning capability of intelligent traffic systems. However, traffic flow prediction is a very challenging problem because the spatial–temporal correlation among roads is complex and changeable. Most of the existing methods do not reasonably analyze the dynamic spatial–temporal correlation caused by the changing relationship of traffic patterns among roads, thus cannot get satisfactory results in the medium and long-term traffic prediction. To address these issues, a novel Multisensor Data Correlation Graph Convolution Network model, named MDCGCN, is proposed in this paper. The MDCGCN model consists of three parts: recent, daily period and weekly period components, and each of which consists of two parts: 1) benchmark adaptive mechanism and 2) multisensor data correlation convolution block. The first part can eliminate the differences among the periodic data and effectively improve the quality of data input. The second part can effectively capture the dynamic temporal and spatial correlation caused by the changing relationship of traffic patterns among roads. Through substantial experiments conducted on two real data sets, results indicate that the proposed MDCGCN model can significantly improve the medium and long-term prediction accuracy for traffic networks of different sizes, and is superior to existing prediction methods.

1. Introduction

Accurate and real-time traffic flow prediction is one of the most critical tasks in intelligent transportation systems (ITS) [1] and of great importance to traffic managers and travelers [2]. Knowing reliable traffic information (such as traffic congestion, traffic volume, etc.) in advance, can help traffic managers better formulate and implement traffic planning strategies, improve the operational efficiency of traffic networks, alleviate traffic congestion, and effectively reduce public safety risks. At the same time, it can help travelers better plan their travel routes, reducing time costs and economic losses. Therefore, traffic flow prediction has become an indispensable part of urban life and has attracted much attention.

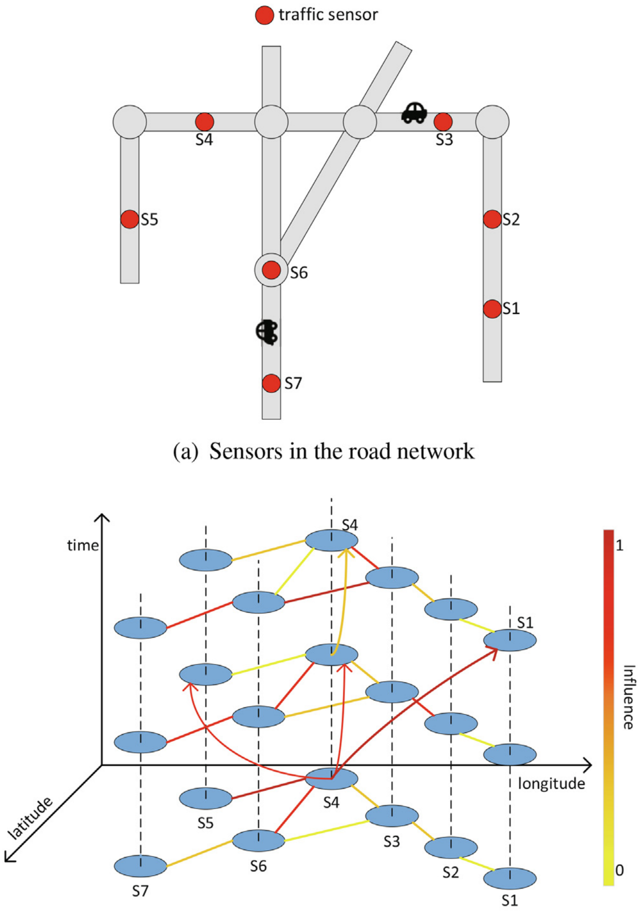
Actually, in order to improve the accuracy of traffic prediction, researchers have made a lot of attempts. Traditional statistical methods were first applied to traffic flow prediction problems, such as Autoregressive Integral Moving Average (ARIMA) [3]. This type of model is only suitable for relatively stable and linearly changeable traffic flow prediction, which cannot meet the actual Application requirements [4]. After that, traditional machine learning models, such as Support Vector Machine (SVM) [5], Support Vector Regression Machine (SVR) [6,7], Bayesian method [8] and K-nearest neighbor [9], were utilized to process highly non-linear data in traffic flow prediction, but their prediction performance relays on careful feature engineering, so these models still cannot be applied to the spatio-temporal correlation analysis of traffic flow data. In recent years, deep learning methods have been applied to traffic prediction problems. For examples, a Graph Convolutional Network (GCN) [10] is used to effectively extract the spatial features of traffic topology network, and a Convolutional Neural Network (CNN) [11] is used to process temporal features with adding the periodicity of the traffic data to improve the traffic flow prediction performance.

Although the above studies have achieved good results in the short term (5–15 min) traffic prediction, most of the existing methods still cannot make satisfactory progress in the medium to long term (15 min–1 h) traffic flow prediction, mainly due to the following challenges. 1) Sensitive periodic data. The periodicity of traffic flow data is greatly affected by people's daily life, which leads to

[*] Corresponding author.
E-mail addresses: 9109@ahu.edu.cn (W. Li), zhangyiwen@ahu.edu.cn (Y. Zhang).

https://doi.org/10.1016/j.neucom.2020.11.032
0925-2312/© 2020 Elsevier B.V. All rights reserved.

the difference among data periodic features. If improper data is selected as the input of the model, the prediction accuracy of the model will decrease. Therefore, how to reasonably analyze the data periodic and select the appropriate data makes the prediction of the distant future extremely challenging. (2) Complex traffic pattern changes. Fig. 1(b) demonstrates the changing relationship of traffic patterns among roads. The seven nodes (S1–S7) in the spatial dimension represent the highway network structure composed of sensors, while the three time slices in the temporal dimension represent the current road network structure at each moment. Solid lines between node pairs represent the degree of correlation of traffic patterns, and the darker the color, the higher the correlation. In short-term spatial–temporal dimension, sensor S4 is closely associated with its traffic patterns of the next moment, at the same time the nearby road (S5, S6 and S3) also make influence, however, this influence changes dynamically. For example, when S4 is located in residential areas, S4 is highly correlated with traffic patterns on nearby roads during peak commuting period, while its correlation with nearby roads decreases during non-commuting period. In the long-term spatial–temporal dimension, the traffic flow at the current moment at S4 will not only has a long-term impact on the traffic patterns at the future moment and at the same location, but also has an impact on the future moments of other roads with the traffic flow on the road network (for example, when there is a traffic jam at S4). In Fig. 1(b) when a special event occurs at S1 (such as the big games), many people from S4 (residential area) will gather at S1 and this will continue for a long time. During this period, two road traffic patterns with strong similarity and highly correlated, but this kind of correlation between roads also dynamically changes, so it is often hard to get effective analysis. Therefore, it is a challenging problem to reasonably analyze the changing relationship of traffic patterns among different roads and effectively capture the temporal and spatial correlation among them to predict the long-term traffic conditions.



(a) Sensors in the road network

(b) Dynamic spatial correlations

Fig. 1. Complex spatio-temporal correlations.

In response to the above challenges, we propose a traffic flow prediction method based on MDCGCN to predict future traffic data. This model can reasonably analyze the periodicity of traffic data in a data-driven way and select the periodic data highly correlated with the current moment as the input. In addition, it can also analyze the changing relationship of traffic patterns among roads according to the traffic condition of the current period, and effectively capture the dynamic spatial–temporal correlation among them. Our major contributions are summarized as follows:

- We propose a benchmark adaptive mechanism to eliminate potential differences among periodic data. Specifically, the benchmark adaptive mechanism automatically selects the data highly related to the current time series from the time series of multiple daily period as the input of the component, improving the overall prediction performance of the proposed model.
- We propose a novel multivariate data associative convolution block, which aims to dynamically capture the spatio-temporal correlation caused by traffic pattern changes between roads in a data-driven way, effectively improving the model's prediction performance in the medium and long term.
- We perform extensive experiments on real traffic datasets. These experimental results show that the proposed model is significantly better than the existing time series models and graph convolution models in the medium and long-term prediction effect.

2. Literature review

To improve the traffic planning capability of ITS, the problem of traffic prediction has attracted widespread attention of researchers, and achieved good results through continuous research and practice [12]. Generally speaking, traffic prediction methods are roughly divided into three categories: traditional statistical methods, traditional machine learning methods, and deep learning methods.

In the traditional statistical methods, the Historical Average method (HA) and Autoregressive Integrated Moving Average (ARIMA) [3] models have been widely used. Subsequently, the variants of the ARIMA model, such as Combining Kohonen maps with ARIMA time series models(KARIMA) [13], Seasonal Autoregressive Integrated Moving Average (SARIMA) [14], etc. were proposed to improve the model's prediction accuracy, but statistical methods have limited processing capabilities for complex and nonlinear traffic data [15].

Compared with the above methods, traditional machine learning methods can model more complex data. For examples, SVM [5,16,17] and SVR [6,18] can map low-dimensional non-linear data to high-dimensional space through kernel functions, so as to analyze data features for prediction. However, the choice of kernel functions has become a key factor to affect model performance. Besides, Bayesian methods [8], K nearest neighbors [9,19,20] and Artificial Neural Networks (ANN) [21] have also been used for traffic flow prediction. Although traditional machine learning methods have a solid mathematical theoretical foundation, these methods rely on feature engineering and expert experience [22]. Consequently, for these methods, it is difficult to effectively improve the accuracy of the model when processing complex, highly nonlinear data [4,23].

At present, deep learning methods have achieved rich results in the domains of image processing and natural language processing [24]. More and more researchers are applying deep learning to the mining of traffic data with temporal and spatial correlation. Huang et al. [25] and Lv et al. [26] used Deep Belief Networks (DBN) and Stacked Autoencoder Models (SAEs) to deepen the network layers for learning the features in traffic data. Then koesdwiady et al. [27] combined the traffic information with the weather information to further improve the prediction performance of the DBN model. But, none of the above models can analyze the temporal correlation in traffic data well. To capture the long-term temporal correlation of traffic data, Long Short-Term Memory (LSTM) [28–31] Gated Recurrent Unit Network (GRU) [32] and Nonlinear Autoregressive With External Input (NARX) [29] models were employed to improve the long-term prediction accuracy of traffic flow, however, these methods ignore the spatial correlation in the traffic network structure. As Convolutional Neural Networks (CNNs) [11,33,34] have made significant breakthroughs in the domain of vision, researchers also strive to apply CNN to the domain of traffic data prediction for capturing local spatial features. Hence, Zhang et al. [35] proposed a Deep Spatio-Temporal Residual Networks (ST-ResNet) to predict the flow of people.

Although traditional convolutional neural networks can handle local spatial features, this type of method limits the input to standard grid data. However the Graph Convolution Network (GCN) [10,36–38] can convolve graph-structured data, so it is widely used in spatio-temporal correlation analysis of processing traffic flow data. But two mainstream methods (spatial method and spectral method) mainly rely on graph adjacency matrix to deal with spatial–temporal correlation, so how to construct the relationship between nodes in the graph is the core of these methods. Seo et al. [39] proposed Graph Convolution Recurrent Network (GCRN), which combines graph convolution with a recurrent neural network. Yu et al. [40] proposed a Graph Convolutional Neural Network with a Gating Mechanism (GLU-STGCN), which avoids the complex structure of recurrent neural networks and retains the ability to capture long-term time correlation. Guo et al. [41] proposed a novel Attention Based Spatial–Temporal Graph Convolutional Network (ASTGCN), which combined attention mechanisms to resolve heterogeneity, and considered the periodicity of traffic data for traffic prediction. However, when considering the data periodic, these models simply select the data in chronological order as the input of the model, ignoring the differences brought by the periodic data due to people's daily activities. In addition, these models only consider the connectivity of the roads and capture the spatial–temporal correlation among the target road and all its neighbors in a fixed range, but ignore the spatial–temporal correlation caused by the changing relationship of traffic patterns among the roads.

Motivated by the studies mentioned above, considering the differences among the periodic data and the changing relationship of the traffic patterns, a novel network structure named multisensor data correlation graph convolution network (MDCGCN) is proposed to predict traffic flow.

3. Preliminary

In this section, we give definitions of related concepts in the paper.

Definition 3.1. (*Traffic topology network*): The topological structure of the traffic topology network is defined as a weighted undirected graph $G = (V, E)$. In this paper, the road section covered by the road sensor is used as a node, where V is the node set, $V = \{v_1, v_2, \ldots v_N\}$, and N is the number of node. E is the set of connected edges of the traffic network, indicating whether there is a real connection between the nodes. The adjacency matrix of a graph G is expressed as $A^d \in R^{N\times N}$,which can be calculated by.

$$A^d_{i,j} = A^d_{j,i} = \begin{cases} \frac{1}{d_{i,j}}, & d_{i,j} \neq 0 \\ 0, & d_{i,j} = 0 \end{cases} \tag{1}$$

The elements in the adjacency matrix only include 0 and $\frac{1}{d_{i,j}}$, where 0 denotes there is no connectivity between nodes, and $\frac{1}{d_{i,j}}$ represents connectivity between nodes, where $d_{i,j}$ indicates the actual distance between nodes. Obviously, the farther the distance, the smaller the weight, and the closer the distance, the greater the weight.

Definition 3.2. (*Feature matrix*): $X \in R^{N\times F\times T}$. We take the traffic information in the network as the attribute feature of the node, expressed as $X \in R^{N\times F\times T}$, where T is the length of historical time in the past, and F is the number of road condition features. $X_t \in R^{N\times i\times t}$ is represented as the i-th road condition feature at time t. Therefore, each node can obtain a feature vector of length p at each time point.

Definition 3.3. (*Traffic flow prediction*): The goal of traffic flow prediction is to know the traffic condition information $\{X_t | t = 0, 1, \ldots, n\}$ in the past period and predict the traffic condition of $X_{n+\Delta t} \in R^{N\times \Delta t}$ in the future $(n + \Delta t)$ time interval, where n is the time interval at the last observed of the historical time series, and Δt is the future period that needs to be predicted after the last moment.

4. Multisensor data correlation graph convolution network model

Fig. 2 shows the multisensor data correlation graph convolutional network (MDCGCN) model proposed in this paper. The model is mainly composed of three parts, which make use of three attributes of traffic data: recent, daily period and weekly period components. In MDCGCN model, the daily period component includes a layer of benchmark adaptive mechanism (BA-Block). In addition, the three components also include two layer multisensor data correlation convolution modules, 2D standard convolution and multi-component gated fusion.

4.1. Data preprocessing

As the fact that traffic data has a certain period at the macro level [41], considering the data periodicity can improve the prediction accuracy of the model. We use the daily periodic data as the input of the daily period component. However, the slight difference still exists in the traffic flow changes of the daily period, especially the difference between the traffic flow changing trends on weekends and workdays. Besides, by observing the data changing trends on the same day of each week, it can be found that the characteristics of the weekly periodic exist in the data. Hence, in order to make up for the shortcomings of daily periodic, we use weekly periodic data as the input of the weekly period component. Note, in this paper, we use the same slicing method as [41], with a 1-day interval as a daily period and a 7-day interval as a weekly period.

Suppose that the sampling frequency is p times a day, that is, the time series length p is obtained every day. Also, assume that the current time is T_0, and the length of the time window to be predicted is T_p. As an example for showing each component input, we

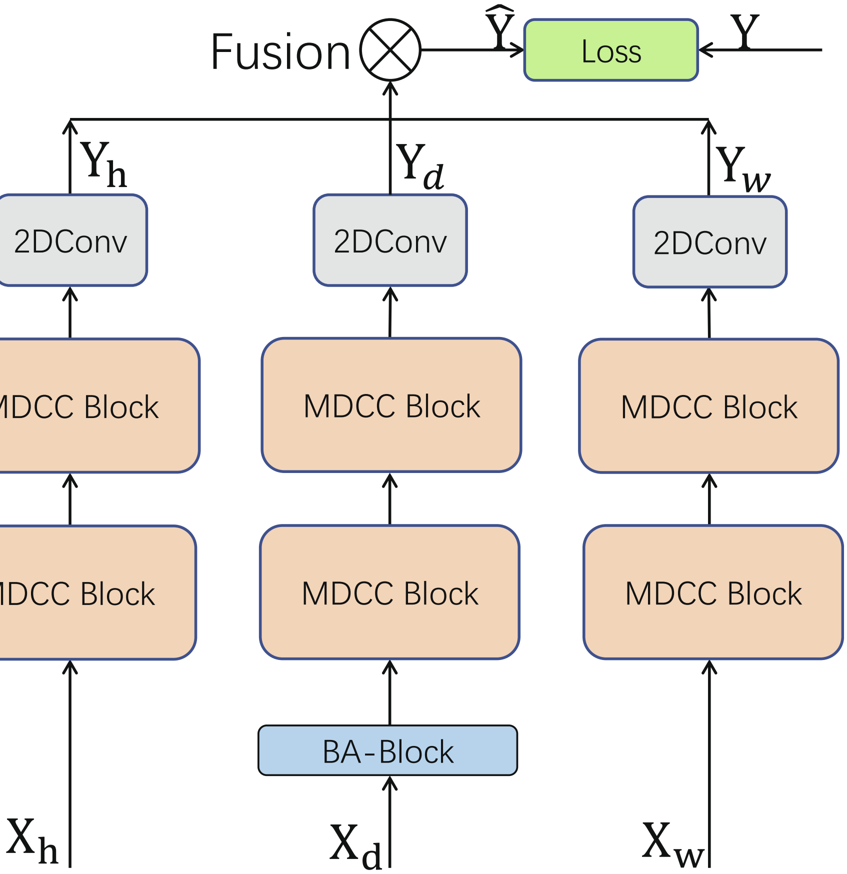


Fig. 2. The architecture of MDCGCN.

intercept the length of T_h, T_d, T_w, and three-time slices along the past time axis as the input of the model's recent, daily, and weekly components in Fig. 3, where T_h, T_d, and T_w are 2 times, 7 times, and 2 times T_p, respectively. The specific interception method of the three-time slices are as follows:

(1) Recent data: $X_h = \left(x_{t_0-T_h+1}, x_{t_0-T_h+2}, \ldots, x_{t_0}\right) \in R^{N\times F\times T'_h}$, that is, the history data series adjacent to the predicted interval, which is shown with green part in Fig. 3, where $T'_h = T_h \ * T_p$. Intuitively, traffic congestion is formed gradually with the change of time. Therefore, the data changing trends of the previous time will always affect the data changes of the next time.

(2) Daily periodic data: $X_d = (x_1, \ldots, x_i) \in R^{N\times F\times T'_d}$, where $x_i = \left(x_{t_0-\left(\frac{T_d}{T_p}-i-1\right)*\ p+1}, \ldots, x_{t_0-\left(\frac{T_d}{T_p}-i-1\right)*\ p+T_p}\right)$ and $T'_d = T_d \ * T_p$. Each day in the week has the same time as the predicted target. Also, the time slice composition of the segment is shown with the blue part in Fig. 3. Due to the influence of the regularity of people's daily life, the traffic flow data of the daily commuting peak show a strong similarity. The daily period component considers the period caused by human factors and provides additional auxiliary information for prediction.

(3) Weekly periodic data: $X_w = (x_1, \ldots, x_i) \in R^{N\times F\times T'_w}$, where $x_i = \left(x_{t_0-\left(\frac{T_w}{T_p}-i-1\right)*\ p+1}, \ldots, x_{t_0-\left(\frac{T_w}{T_p}-i-1\right)*\ p+T_p}\right)$ and $T'_w = T_w \ * T_p$, where T_w has the same week as the predicted target. Also, attributes and time slices of the same period are shown with red part in Fig. 3. Similarly, the traffic patterns on the same day of each week is periodic. For example, the traffic patterns on Saturday has a strong similarity with the traffic condition on Saturday in history, but it is very different from the traffic condition on weekdays. Therefore, we consider the weekly period of the data to make up for the deficiency of the daily period.

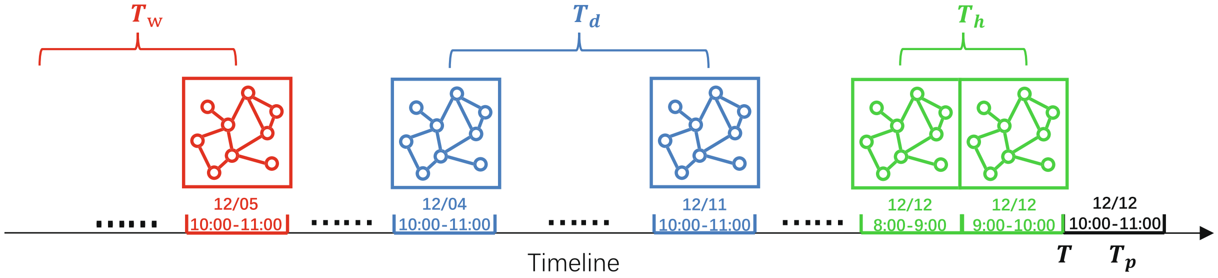


Fig. 3. Example of each component input.

4.2. Benchmark adaptive mechanism

Because the traffic patterns of the same period in the daily period data are not always highly related, for example, a certain group of people go to work on Monday, but not on Tuesday and Wednesday, then this routine will affect the traffic condition of each day. As a result, in this paper, benchmark adaptative mechanism is added to preprocess the time slices input by the daily period component, to deal with the differences among traffic modes in daily period. The specific process is as follows: we select the data of the previous hour at the current time t_0 as the key time slice, and let $X_h = \left(x_{t_0-\frac{p}{24}+1}, x_{t_0-\frac{p}{24}+2}, \ldots, x_{t_0}\right) \in R^{N\times F\times \frac{p}{24}}$, where p is the length of the time series data obtained each day with 24 h. According to the following equations:

$$r = \frac{Z\sum x_h x_i - \sum x_h \sum x_i}{\sqrt{Z\sum x_h^2 - \left(\sum x_h\right)^2}\sqrt{Z\sum x_i^2 - \left(\sum x_i\right)^2}} \tag{2}$$

We can calculate each x_i in $X_d = (x_1, x_2 \ldots, x_i) \in R^{N\times F\times T_p\times D}$ to get $r_d = (r_1, r_2 \ldots, r_i) \in R^{N\times F\times T_p\times D}$. Besides, the N_d time slices are selected with the highest correlation coefficient with X_h and are added to the data series of the daily periodic. In addition, we use $X_d = (x_1, x_2, \ldots, x_i) \in R^{N\times F\times T'_d}$ as the input of the network, where $T'_d = N_d \ * T_p$, and Z is the number of data in X_h and X_d(X_h and X_d have the same amount of data).

4.3. Multisensor data correlation convolution block

Fig. 4 shows a multisensor data correlation convolution block, which includes four sub-modules: dynamic graph generation module, temporal attention module, spatial attention module, bigroup graph convolution module, and 2D temporal convolution module.

4.3.1. Temporal attention module

In the temporal dimension, with the passage of time and the development of things, the traffic patterns at each time point in the past will have different degrees of influence on the future. At the same time, in the historical time series, under different circumstances, the correlation of traffic patterns between different periods also shows the differences. Hence, to capture the dynamic temporal correlation between each period, the temporal attention module is introduced and can be calculated by (3) and (4),

$$E = U_e \cdot \sigma\left(\left(\left(X^{(l-1)}\right)^T W_1\right) W_2 \left(W_3 X^{(l-1)}\right) + b_e\right) \tag{3}$$

$$E'_{i,j} = \frac{exp\left(E_{i,j}\right)}{\sum_{j=1}^{T_{l-1}} exp\left(E_{i,j}\right)} \tag{4}$$

where $X^{(l-1)} = \left(x_1, x_2, \ldots, x_{T_{l-1}}\right) \in R^{N\times F_{l-1}\times T_{l-1}}$is the input of the l-th multisensor data correlation convolution block. $W_1 \in R^N, W_2 \in R^{F_{l-1}\times N}, W_3 \in R^{F_{l-1}}, b_e \in R^{1\times T_{l-1}\times T_{l-1}}, U_e \in R^{T_{l-1}\times T_{l-1}}$ are learnable parameters and σ represents the activation function, F_{l-1} is the data feature of the $l-1$ layer output, $F_0 = 3$, and T_{l-1} is the time length of the output data of the $l-1$ layer. E stands for attention score matrix and is determined by dynamically changing inputs. $E_{i,j}$ represents the degree of influence between the i-th time point and the j-th time point. We multiply the normalized time notice matrix by the input and get $\widetilde{X}^{(l-1)} = \left(\tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_{T_{l-1}}\right) = \left(x_1, x_2, \ldots, x_{T_{l-1}}\right)E'$. After that, $\widetilde{X}^{(l-1)} \in R^{N\times F_{l-1}\times T_{l-1}}$ will be input into the spatial attention module to calculate the dynamic spatial correlation.

4.3.2. Spatial attention module

In the spatial dimension, the traffic conditions among different roads affect each other and change with changes in geographical location and time. Likewise, considering the dynamic spatial correlation among roads, we introduce an attention mechanism [11]. According to the current input, the spatial correlation among the roads can be adjusted adaptively. Taking the recent component input as an example, the dynamic spatial correlation can be calculated as follows:

$$E = V_e \cdot \sigma\left(\left(\left(X_h^{(l-1)}\right)^T U_1\right) U_2 \left(U_3 X_h^{(l-1)}\right) + b_e\right) \tag{5}$$

$$S'_{i,j} = \frac{exp\left(S_{i,j}\right)}{\sum_{j=1}^{N} exp\left(S_{i,j}\right)} \tag{6}$$

where $U_1 \in R^{T_{l-1}}, U_2 \in R^{F_{l-1}\times T_{l-1}}, U_3 \in R^{F_{l-1}}, V_e \in R^{N\times N}, b_e \in R^{N\times N}$ are learnable parameters and $\widetilde{X}_h^{(l-1)} = \left(\widetilde{X}_1, \widetilde{X}_2, \ldots, \widetilde{X}_{T_{l-1}}\right) \in R^{N\times F_{l-1}\times T_{l-1}}$. F_{l-1} is the data feature of the l-th layer multisensor data correlation convolution block output, $F_0 = 3$, and T_{l-1} is the time length of the output data of the l layer. When $l = 1$, the recent module $T_0 = T'_h$, the daily period component and weekly period component are T'_d, T'_w, respectively. The $S'_{i,j}$ matrix indicates the degree of influence of node j on node i. In the process of graph convolution, the spatio-temporal attention matrix S' is multiplied to dynamically adjust the influence of time and space dimensions between nodes.

4.3.3. Dynamic graph generation module

The spatial–temporal correlation between roads should consider not only distance and connectivity, but also the changing relationship among traffic patterns. In this case, it is far from enough to capture only the temporal and spatial correlation of traffic topology networks. Therefore, we propose a dynamic graph generation module to construct a real-time traffic pattern relationship graph.

The traffic pattern relationship graph breaks the limitations of the traffic topology network structure (constrained by the distance between roads) and makes the connections among roads with similar traffic patterns closer. To illustrate, Fig. 5 shows the structure comparison of the traffic topology network and traffic pattern relationship graph. Taking S4 as an example, the Fig. 5(a) shows two results in normal situation. The first is, when two roads is in the short distance, the degree of interaction gets higher with the assigned weight gets higher. On the contrary, the second tells that the degree and weight become lower when the roads are in long distance. However, for complex traffic topology network, each road is affected not only by neighboring roads but also by roads with similar traffic patterns. Such as, when traffic congestion occurs during the rush hour, the traffic of S4 will always have a huge impact on the road in the direction of S1 within 1–2 h, however, the impact on S3 is small. So, S5, S1, S7, and S4 have highly similar traffic patterns. In Fig. 5(b), S5, S1, and S7 will become first-order neighborhoods of the S4, which aims to capture the spatio-temiral correlation among them. Therefore, to compensate for the shortcomings that the traffic topology network only captures the spatio-temporal correlation on limited neighbors, we analyze the relationship among road traffic patterns to build dynamic traffic pattern relationships graph. Next, we give the specific calculation process of the dynamic graph generation module.

As the average speed is an important factor to reflect traffic congestion and accidents, we use the average speed of vehicles as an important indicator to measure the relationship of road traffic patterns. The correlation between the speed changes of each road are calculated using pearson correlation coefficient, according to the following equations:

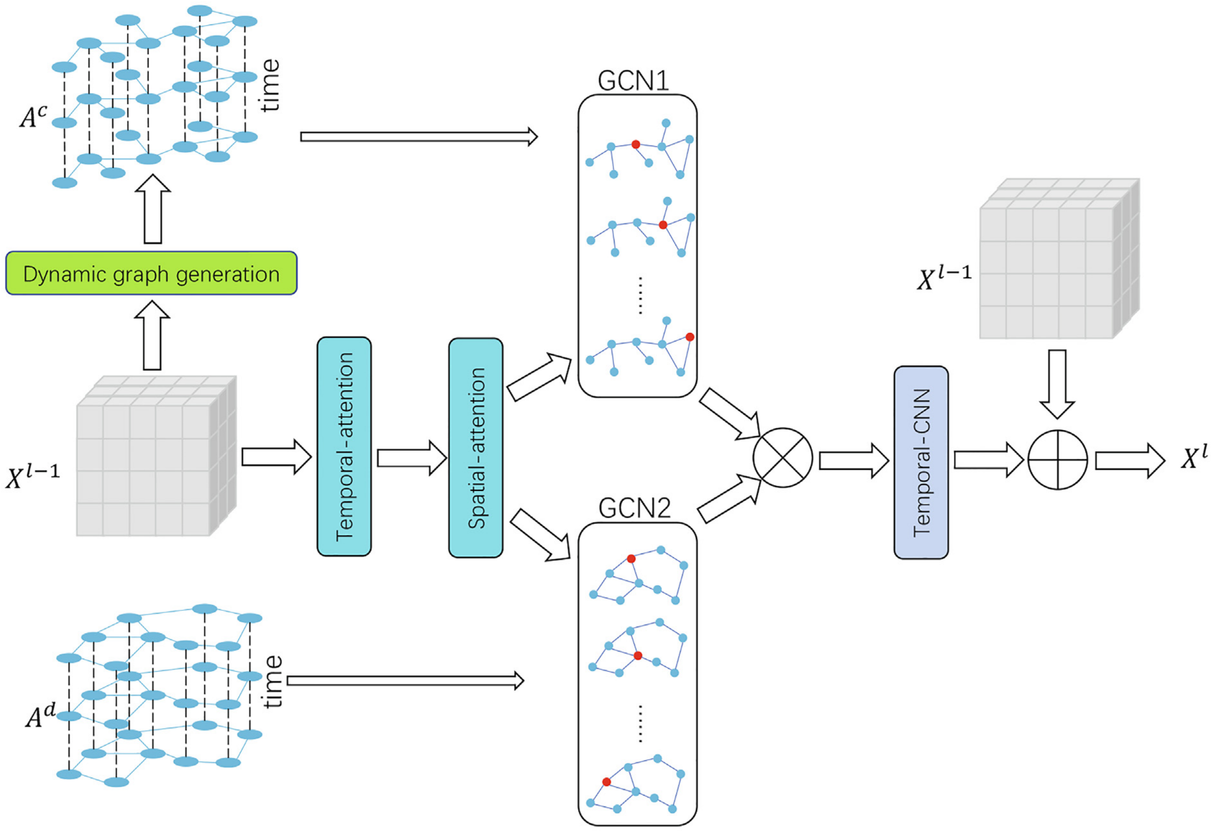
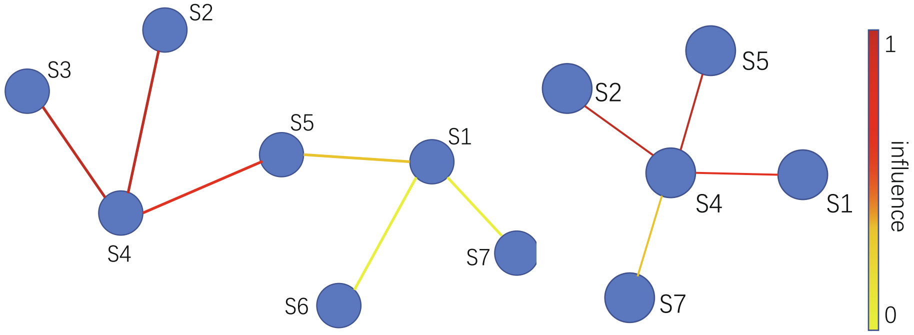


Fig. 4. Multisensor Data Correlation Convolution Block (MDCC Block).



(a) traffic topology network

(b) traffic pattern relationship graph

Fig. 5. Example of graph generation module.

$$p_{i,j} = \frac{Z\sum x_i^{l-1}x_j^{l-1} - \sum x_i^{l-1}\sum x_j^{l-1}}{\sqrt{Z\sum\left(x_i^{l-1}\right)^2 - \left(\sum x_i^{l-1}\right)^2}\sqrt{Z\sum\left(x_j^{l-1}\right)^2 - \left(\sum x_j^{l-1}\right)^2}} \tag{7}$$

$$A^c_{i,j} = A^c_{j,i} = \begin{cases} p_{i,j}, & p_{i,j} \neq 0 \\ 0, & p_{i,j} = 0 \end{cases} \tag{8}$$

In this paper, we use pearson correlation to measure the correlation between the traffic patterns of the two road sections, and the correlation coefficient is used as the weight connecting the two nodes. Here, the traffic pattern relationship graph is expressed as $G_c(V,E)$, where A^c represents the adjacency matrix of the traffic pattern relationship graph, $p_{i,j}$ represents the correlation between node i and node j, x_i^{l-1} and x_j^{l-1} represent the time series values of node i and node j in the l-th layer. Furthermore, we make each node have the same data volume as Z.

4.3.4. Bigroup graph convolution module

In this study, the traffic network is naturally regarded as a graph structure, and we consider roads as the node on the graph and the traffic information as the signal on the graph [42]. Hence, in order to make full use of the topological properties of the two weighted spatial graphs $G_d(V,E)$ and $G_c(V,E)$, at each time slice of the spatial graph, we use graph convolution based on spectral graph theory to process signals, capturing the relationships among signals on $G_d(V,E)$ and $G_c(V,E)$ in the spatial dimension, respectively. In spectral graph theory, the graph is represented by the corresponding

Laplacian matrix, and the properties of the graph are studied by analyzing the eigenvalues and eigenvectors of the Laplacian matrix of the graph, where the Laplacian matrix of the graph is usually expressed as $L = D - I$. In this paper, we use the normalized form that is $L = I_N - D^{-\frac{1}{2}}AD^{-\frac{1}{2}} \in R^{N\times N}$, where A is the adjacency matrix, and I_N is the identity matrix, and $D \in R^{N\times N}$ is the diagonal matrix of the sum of the degrees of each node, and $D_{i,i} = \sum_j A_{i,j}$. Laplace matrix is spectrally decomposed as $L = U\Lambda U^T$, where $\Lambda = diag\left(\left[\lambda_0, \ldots, \lambda_{N-1)}\right]\right) \in R^{N\times N}$ is determined by the eigenvalues of the Laplacian matrix that is a diagonal matrix and U is the Fourier basis[43]. The convolution operation is usually expressed as a Eq. (9), where $g_\theta(\Lambda)$ is expressed as a convolution kernel, and σ represents the activation function.

$$y_{output} = \sigma\left(Ug_\theta U^T x\right) \tag{9}$$

Since the Fourier change of the convolution kernel $g_\theta(\Lambda)$ is performed for each convolution. When the scale of the graph is very large, the calculation is expensive and the time complexity is $O(n^2)$. To reduce the time complexity, we use Chebyshev polynomials to fit the convolution kernel [31], which can be calculated as

$$g_\theta(\Lambda) = \sum_{k=0}^{K-1} \beta_k T_k\left(\tilde{\Lambda}\right) \tag{10}$$

where $\beta_k \in R^k$ is the Chebyshev polynomial coefficient, and $\tilde{\Lambda} = 2\Lambda/\lambda_{max} - I$ is expressed as the maximum eigenvalue of the Laplacian matrix. Based on Chebyshev polynomial properties, the recursive formula is defined as $T_k(x) = 2T_{k-1}(x) - T_{k-2}(x)$, where $T_0 = 1, T_1 = x$. By Chebyshev's solution, we use the convolution kernel with each node as the center to extract the influence of the features of the $k-1$ order neighborhood around the node on its central node, and finally use the linear correction unit (ReLU) as the activation function, which is $ReLU(g_\theta *_G x)$. To dynamically adjust the degree of influence between each node, each $T_k\left(\tilde{\Lambda}\right)$ is multiplied by the spatiotemporal attention matrix $S' \in R^{N\times N}$, namely $T_k\left(\tilde{\Lambda}\right) \odot S'$, where $\odot$ is a Hadamard product. Therefore, the final convolution formula can be expressed as

$$g_\theta(\Lambda)x = \sum_{k=0}^{K-1} \theta_k \left(T_k\left(\tilde{\Lambda}\right) \odot S'\right)x \tag{11}$$

In this module, the above-mentioned graph convolution operation is performed on the constructed two graph structures. The feature map obtained after a convolution is weighted and summed by Eqs. (12) and (13) to obtain the final feature graph as the input of the next module.

$$\hat{F} = \left((g_\theta^d(\Lambda) + g_\theta^c(\Lambda))X^{l-1}\right) \odot W \tag{12}$$

$$(g_\theta^d(\Lambda) + g_\theta^c(\Lambda)) = \left(\sum_{k=0}^{K-1} \theta_k \left(T_k^d\left(\tilde{\Lambda}\right) + T_k^c\left(\tilde{\Lambda}\right)\right) \odot S'\right) \tag{13}$$

where $T_k^d\left(\tilde{\Lambda}\right)$ and $T_k^c\left(\tilde{\Lambda}\right)$ represent the convolution kernels of original traffic topology network and the traffic pattern relationship graph, respectively. $W \in R^{m\times m}$ is a learnable parameter and m is number of convolution kernel.

4.3.5. Temporal convolution module

The Long Short-Term Memory (LSTM) [44] can capture long-term temporal correlation through a gating mechanism and has been widely used in time series analysis models. However, because the gating mechanism is too complicated, this increases the complexity of the model while increasing the input of time series data, thus reducing the final prediction accuracy of the model. Conversely, Convolutional Neural Network (CNN) has the advantages of simple structure and no restriction from the previous step. Inspired by [45], we use convolutional neural networks to extract features in the temporal dimension. To avoid the loss of original data during the convolution process, a residual block is added. Hence, the temporal correlation can be captured by

$$X^l = relu\left(\hat{F} * W + b + \left(X^{l-1} * V + c\right)\right) \tag{14}$$

where $W \in R^{k\times m\times n}$ and $V \in R^{k\times m\times n}$ represent different convolution kernels, k and m are the length and width of the convolution kernel, and n represents the number of channels. $b \in R^n, c \in R^n$ are the paranoid vectors corresponding to the convolution kernel. Besides, W, b, V, and c are learnable parameters, X^{l-1} and X^l represent input and output of the l-th layer MDCC block, respectively.

Finally, a 2D convolution layer is added to obtain the predicted values of each component. Taking the recent component output as an example, the calculation equation is as follows:

$$Y_h = X_h^l * W^h \tag{15}$$

where $W^h \in R^{1\times 1\times T_p}$ are convolution kernel parameters.

4.4. Multi-component fusion module

In this section, we use Eq. (16) to fuse the predicted values of the recent, the daily period and weekly period components to get the final output, so that the model can adaptively decide which component's results are more important.

$$\hat{Y} = W_h \odot Y_h + W_d \odot Y_d + W_w \odot Y_w \tag{16}$$

where $\odot$ is a Hadamard product, $W_h \in R^{T_p\times T_p}, W_d \in R^{T_p\times T_p}$, and $W_w \in R^{T_p\times T_p}$ are learnable parameters.

5. Experiment

To evaluate the performance of the model, we performed comparison experiments on two real highway traffic datasets.

5.1. Data description

In this work, we used two real-world datasets [41] (real highway datasets PEMSD4 and PEMSD8 in California, USA) to evaluate our model. (1) PEMSD4 includes 307 detectors, and the selected time range is from January to February 2018. (2) PEMSD8 includes 170 detectors, and the selected time range is from July to August 2016. In both data sets, the sample collection interval was 5 min. Therefore, each node on each road collects 288 pieces of data every day, and each piece of data includes two characteristics of volume and speed. We use the first 80% of the data strictly in chronological order as the training set, the middle 10% as the verification set, and the last 10% as the test set. The missing values are filled by linear interpolation. In addition, the data is normalized to zero mean $x' = x - mean(x)$, so that the average value is 0.

5.2. Experimental parameter settings

In order to evaluate the proposed MDCGCN model and achieve the best performance, we followed the previous work [41], carried out manual parameter adjustment after many tests, and finally determined the experimental parameter setting as follows:1) we use $T_h = 2, N_d = 1$, and $T_w = 2$ as history data time step, $T_p = 12$ as the future time step to be predicted, 2) all graph convolution layers use 64 convolution kernels with a kernel size of $K = 3$, 3) the number of channels for all temporal convolutional layers in the MDCC Block is set to 64 and the time span of the temporal convolutional is controlled by adjusting the step size of the traffic flow data, 4) the mean square error (MSE) of the predicted value and the real value is used as a loss function to minimize it by backpropagation, 5) in the training phase, the batch size is set to 16 and the learning rate is set to 0.0001.

5.3. Baseline method

To ensure the comprehensiveness and validity of the experiment, we will compare our model with the following 10-time series prediction methods:

(1) **HA**: Here, the average value of the 12 time slices we used aims to predict the next value.
(2) **ARIMA** [3]: Autoregressive integrated moving average is a traditional time series prediction method.
(3) **VAR** [6]: Vector autoregressive model is a time series model, which can capture the relationship between the time series data.
(4) **LSTM** [44]: Long-short-term memory network, which can capture long-term time correlation,is a variant of RNN model.
(5) **GRU** [46]: Gated Recurrent Unit network, which effectively reduces the time complexity of the LSTM model, is a variant of LSTM model.
(6) **GLU-STGCN** [40]: A spatio-temporal graph convolution model includes gating mechanism.
(7) **ASTGCN** [41]: A spatio-temporal graph convolution model, which has attention mechanism, can capture the dynamic spatio-temporal correlation in the road network.
(8) (Spatial–Temporal Graph Convolution)**STGCN** [47]: It is a spatio-temporal prediction model based on convolution sequence layer and graph convolution layer.
(9) (Traffic Graph Convolutional Long Short-Term Memory Neural Network)**TGC-LSTM** [48]: A traffic prediction model combining graph convolution and LSTM network.
(10) (Optimized Graph Convolution Recurrent Neural Network) **OGCRNN** [49]: Which reveals the latent relationship among the road segments from the traffic data.

For models ARIMA, VAR, LSTM and GRU, we use the setting suggested by [50]. For models STGCN, GLU-STGCN, ASTGCN, TGC-LSTM and OGCRNN, we use the default settings from their original proposals.Our implementation is based on MXNet 1.5.1, and they are trained and evaluated on a single NVIDIA Tesla P40 with 24 GB memory.

5.4. Evaluation metrics

In this work, we use two commonly used metrics to evaluate the prediction performance in the domain of traffic flow prediction. These two metrics are mean absolute error (MAE) and root mean squard error (RMSE). Specifically, the calculation equation MAE and RMSE are as follows:

$$MAE = \frac{1}{NT_p}\sum_{j=1}^{N}\sum_{i=1}^{T_p} \mid \hat{Y}_i^j - Y_i^j \mid \tag{17}$$

$$RMSE = \sqrt{\frac{1}{NT_p}\sum_{j=1}^{N}\sum_{i=1}^{T_p}\left(\hat{Y}_i^j - Y_i^j\right)} \tag{18}$$

where $\hat{Y}_i^j$ and Y_i^j represent the predicted and observed traffic flows at the j-th point at time t, respectively. T_P is the length of the time window to be predicted. Generally, lower MAE and RMSE values indicate better model prediction performance.

5.5. Comparison of prediction performance

On PEMSD4 and PEMSD8 datasets, Table 1 shows the performance of our model and baseline method for predicting 15 min, 30 min, 45 min, and 1 h. It is obvious that: 1) Comparing with traditional time series methods, deep learning methods can get better prediction accuracy, which means that deep learning methods can

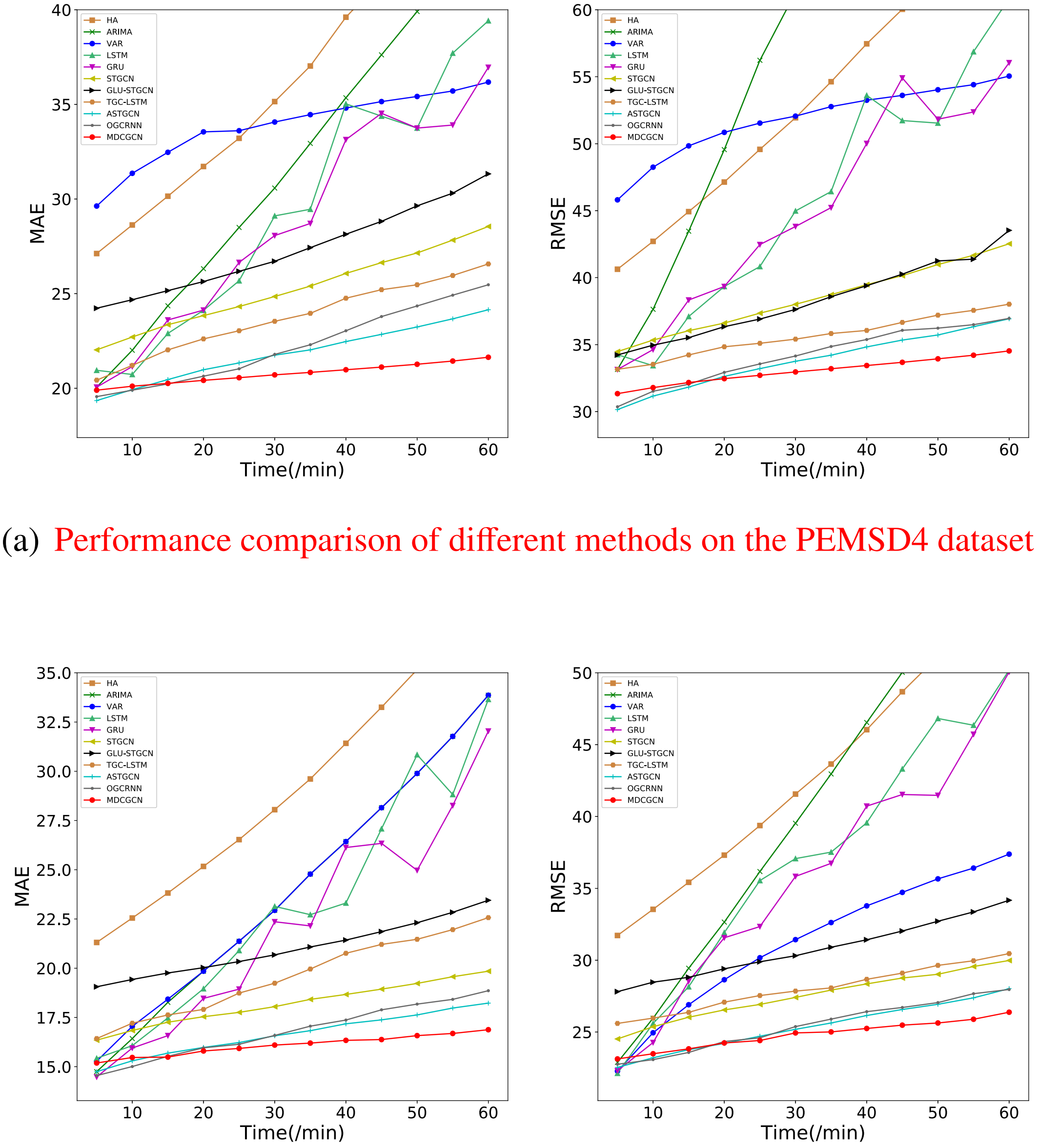
Table 1
The performance comparison of traffic flow prediction methods for PEMSD4 and PEMSD8 datasets. The smaller MAE and RMSE means the better performance, and we bold the best resault.

Data	Method	15 min		30 min		45 min		60 min	
		MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
PEMSD4	HA	30.14	44.93	35.15	51.94	41.87	60.04	54.49	65.49
	ARIMA	24.36	43.47	30.58	61.30	37.62	66.43	44.31	71.02
	VAR	32.47	49.84	34.07	52.06	35.15	53.61	36.18	55.05
	LSTM	22.90	37.10	29.11	44.98	34.38	51.48	39.41	60.84
	GRU	23.61	38.33	28.07	43.82	34.53	54.91	36.96	56.05
	STGCN	23.36	36.04	24.58	38.01	26.64	40.15	28.55	42.53
	GLU-STGCN	25.16	35.52	26.71	37.64	28.82	40.26	31.33	43.53
	TGC-LSTM	22.03	34.23	23.54	35.41	25.21	36.66	26.57	38.02
	ASTGCN	20.46	**31.83**	21.75	33.76	22.82	35.34	24.15	36.93
	OGCRNN	**20.23**	32.04	21.79	34.15	23.79	36.07	25.47	36.96
	MDCGCN	20.26	32.17	**20.17**	**32.96**	**21.12**	**33.68**	**21.64**	**34.53**
PEMSD8	HA	23.84	35.42	28.05	41.65	33.38	48.63	38.47	55.76
	ARIMA	18.25	29.58	22.98	39.61	28.16	50.03	33.87	60.15
	VAR	18.42	26.96	21.47	31.37	23.95	34.83	25.93	37.41
	LSTM	17.46	28.14	23.24	37.05	27.05	43.37	33.67	50.31
	GRU	16.58	28.67	22.35	35.94	26.37	41.47	32.03	50.04
	STGCN	17.16	26.04	18.04	27.48	18.95	28.86	19.55	29.93
	GLU-STGCN	19.73	28.94	20.65	30.36	21.94	32.16	23.59	34.23
	TGC-LSTM	17.63	26.37	19.24	27.85	21.21	29.10	22.57	30.46
	ASTGCN	15.67	23.76	16.59	25.29	17.35	26.43	18.28	28.05
	OGCRNN	15.53	**23.58**	16.59	25.38	17.89	26.71	18.86	27.96
	MDCGCN	**15.49**	23.83	**16.10**	**24.94**	**16.38**	**25.48**	**16.88**	**26.38**

better capture the inherent features in highly complex traffic flow data. 2) Among deep learning methods, models that consider both temporal correlation and traffic network topology (like STGCN, GLU-STGCN, TGC-LSTM, ASTGCN, OGCRNN and our model) are significantly better than traditional deep learning methods (such as LSTM and GRU). Therefore, the experiment shows that it is essential for traffic flow prediction to consider the traffic network structure. 3) Our MDCGCN has obtained the best results on the four prediction intervals compared with the other eight methods. Experiments show that our model can better analyze the spatio-temporal characteristics of traffic flow data.

Fig. 6 shows the trend of prediction accuracy of each model after the prediction interval increases. From the figure, we observe that: 1) Models that only consider temporal correlation, such as HA, ARIMA, LSTM and GRU, can achieve good results in the short term, but with the increase of the prediction interval, the prediction performance of the model decreases rapidly; 2) Due to taking the spatio-temporal correlation into account to some extent, the prediction accuracy of VAR decreases slowly, however, comparing VAR models to predict performance trends on two data sets, we find that the ability to analyze highly complex traffic flow data is still limited; 3) For the deep learning model, STGCN, GLU-STGCN, TGC-LSTM, ASTGCN, OGCRNN and our model consider the road network topology structure and maintain good performance even if the prediction interval is increased. Therefore, road network information plays a vital role in long-term traffic prediction; 4) In contrast, the proposed MDCGCN model has the best performance on both the PEMSD4 and PEMSD8 datasets.

Meanwhile, in order to better compare the performance of our model and the baseline method. Following the suggestions in [51], we apply the Wilcoxon signed-rank test [51] on the MAE and RMSE values of three latest baseline methods and our model. The test results are summarized in Tables 2 and 3 for MDCGCN and the three baseline methods, respectively. It can be observed



(a) Performance comparison of different methods on the PEMSD4 dataset

(b) Performance comparison of different methods on the PEMSD8 dataset

Fig. 6. As the prediction interval increases, the prediction accuracy of different methods changes.

Table 2
Wilcoxon signed-rank tests for RMSE values between MDCGCN and baseline methods on real datasets.

Data set	Prediction interval	MDCGCN vs OGCRNN				MDCGCN vs ASTGCN				MDCGCN vs TGC-LSTM			
		OGCRNN	MDCGCN	Difference	Rank	ASTGCN	MDCGCN	Difference	Rank	TGC-LSTM	MDCGCN	Difference	Rank
PEMSD8	10 min	23.08	23.49	-0.51	-5	23.01	23.49	-0.48	-4	25.97	23.49	+2.48	+4
	20 min	24.34	24.24	+0.10	+1	24.21	24.24	-0.03	-1	27.08	24.24	+2.84	+6
	30 min	25.38	24.94	+0.44	+3	25.19	24.94	+0.25	+3	27.85	24.94	+2.91	+7
	40 min	26.42	25.25	+1.17	+6	26.15	25.25	+0.90	+7	28.67	25.25	+3.42	+9
	50 min	27.05	25.63	+1.42	+8	26.93	25.63	+1.30	+8	29.64	25.63	+4.01	+11
	60 min	27.96	26.38	+1.58	+9	28.02	26.38	+1.64	+10	30.46	26.38	+4.08	+12
PEMSD4	10 min	31.52	21.79	-0.27	-2	31.16	21.79	-0.63	-5	33.53	21.79	+1.74	+1
	20 min	32.93	32.46	+0.47	+4	32.63	32.46	+0.17	+2	34.84	32.46	+2.38	+2
	30 min	34.15	32.96	+1.19	+7	33.76	32.96	+0.80	+6	35.41	32.96	+2.45	+3
	40 min	35.38	33.44	+1.94	+10	34.83	33.44	+1.39	+9	36.06	33.44	+2.62	+5
	50 min	26.23	34.94	+2.29	+11	35.72	33.94	+1.78	+11	37.20	33.94	+3.26	+8
	60 min	26.96	34.53	+2.43	+12	36.93	34.53	+2.40	+12	38.02	34.53	+3.49	+10
			$T = min\{71, 7\} = 7$				$T = min\{68, 10\} = 10$				$T = min\{78, 0\} = 0$		

Table 3
Wilcoxon signed-rank tests for MAE values between MDCGCN and baseline methods on real datasets.

Data set	Prediction interval	MDCGCN vs OGCRNN				MDCGCN vs ASTGCN				MDCGCN vs TGC-LSTM			
		OGCRNN	MDCGCN	Difference	Rank	ASTGCN	MDCGCN	Difference	Rank	TGC-LSTM	MDCGCN	Difference	Rank
PEMSD8	10 min	15.01	15.47	-0.46	-4	15.30	15.47	-0.17	-1	17.21	15.47	+1.74	+2
	20 min	15.96	15.80	+0.16	+1	15.98	15.80	+0.28	+3	17.91	15.80	+2.11	+3
	30 min	16.59	16.10	+0.49	+5	16.57	16.10	+0.47	+4	19.24	16.10	+3.14	+6
	40 min	17.37	16.34	+1.63	+7	17.18	16.34	+0.84	+6	20.76	16.34	+4.42	+9
	50 min	18.18	16.58	+2.40	+9	17.63	16.58	+1.05	+8	21.47	16.58	+4.89	+10
	60 min	18.86	16.88	+3.48	+11	18.23	16.88	+1.35	+11	22.57	16.88	+5.69	+12
PEMSD4	10 min	19.90	20.11	-0.21	-2	19.93	20.11	-0.18	-2	21.21	20.11	+1.10	+1
	20 min	20.64	20.42	+0.22	+3	20.98	20.42	+0.56	+5	22.61	20.42	+2.19	+4
	30 min	21.79	20.71	+1.08	+6	21.75	20.71	+1.04	+7	23.54	20.71	+2.83	+5
	40 min	23.04	20.98	+2.06	+8	22.47	20.98	+1.49	+9	24.76	2.98	+3.78	+7
	50 min	24.35	21.27	+3.83	+10	23.24	21.27	+1.97	+10	25.47	21.27	+4.20	+8
	60 min	25.47	21.64	+3.83	+12	24.15	21.64	+2.51	+12	26.57	21.64	+4.93	+11
			$T = min\{72, 6\} = 6$				$T = min\{75, 3\} = 3$				$T = min\{78, 0\} = 0$		

that the performance of MDCGCN is statistically better than that with OGCRNN, ASTGCN, TGC-LSTM. For example, in RMSE value case of MDCGCN versus OGCRNN (Table 2), OGCRNN is better (negative difference) than MDCGCN for two prediction intervals, while MDCGCN is better (positive difference) than OGCRNN for ten prediction intervals (Table 2). We sum all the positive ranks under the rank column to find R+, which is 71 and similarly, sum all the negative ranks to find R–, which is 7. The minimum of R+ and R– is the T value and it is 7. Since there are 12 prediction intervals, the T value at a significance level of 0.05 should be less than or equal to 13 (the critical value) to reject the null hypothesis [51]. That is, MDCGCN is better than OGCRNN.

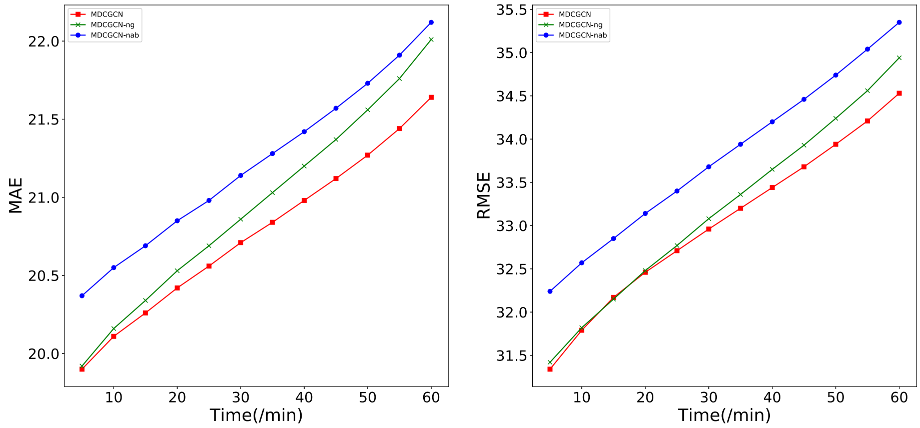
Since our model takes into account the spatial–temporal correlation caused by the changing relationship of traffic pattern among roads, the information gathered at each node is more complex. Therefore, the prediction accuracy of our model compared with other baseline methods was slightly insufficient in the short term, but after 15 min, with the increase of the prediction interval, the advantages of our model became more and more obvious. Furthermore, our model is more suitable for the PEMSD4 dataset, which is more complex than the PEMSD8 dataset in traffic structure, thereby, it is fully verified that our model, considering the spatio-temporal correlation among roads with similar traffic patterns, can effectively improve the long-term prediction performance of complex traffic network structure. We argue that better long-term prediction performance is more suitable for practical applications, because improving long-term forecast accuracy can help road managers and users have enough time to make reasonable traffic planning and travel plans.

Table 4
Introduction to model name.

model	description
MDCGCN	our model.
MDCGCN-ng	The traffic pattern relationship graph is separated in our model.
MDCGCN-nab	The benchmark adaptive mechanism is separated in our model.

5.6. The validity of each module

To verify the function of each part of our proposed module, we evaluate two variants by removing benchmark adaptive mechanism and the traffic pattern relationship graph from MDCGCN separately, which are named as MDCGCN-nab and MDCGCN-ng, respectively, as show in Table 4. Taking the PEMSD4 dataset as an example, Fig. 7 shows the performance indicators predicted by each interval of MDCGCN and the other three models. We can see that MDCGCN is always better than MDCGCN-nab and MDCGCN-ng, which shows the effectiveness of our proposed benchmark adaptive mechanism and the traffic pattern relationship graph in traffic prediction modeling. With the increase of the prediction interval, MDCGCN performs better than MDCGCN-ng, which shows that the proposed the traffic pattern relationship graph effectively improves the long-term prediction of traffic flow. Also, the overall performance of the MDCGCN and MDCGCN-ng models retaining the benchmark adaptive mechanism is better



(a) (b)

Fig. 7. The validity of each module.

Table 5
Introduction to model name.

model	description
MDCGCN	A traffic pattern relationship graph is constructed without scope constraints.
MDCGCN-f	A traffic pattern relationship graph based on first-order neighbors is constructed.
MDCGCN-s	A traffic pattern relationship graph based on sescond-order neighbors is constructed.
MDCGCN-t	A traffic pattern relationship graph based on third-order neighbors is constructed.

than MDCGCN-nab, demonstrating that the proposed benchmark adaptive mechanism improves the quality of the input and is great significant in improving the overall performance of the model.

5.7. Spatio-temporal correlation among roads with long spatial distances

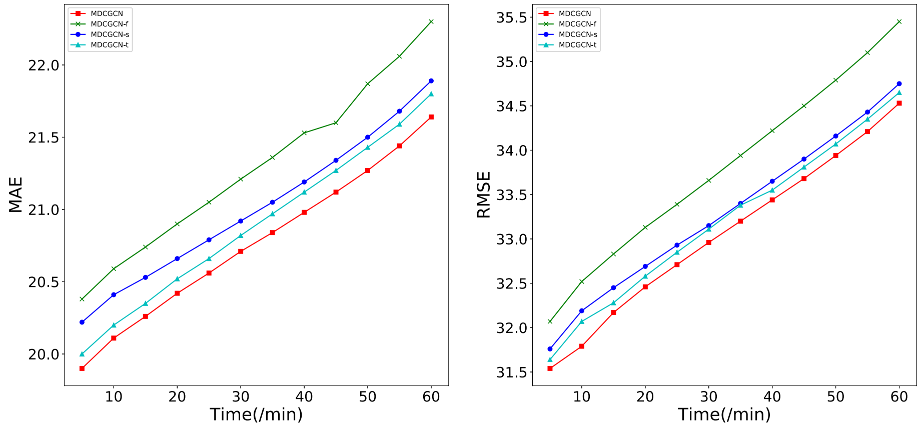
To investigate whether there is a spatio-temporal correlation among roads with similar traffic patterns but could be spatially distant (such as multiple business districts), we evaluated three variants by limiting the MDCGCN model's analysis of traffic pattern relationships in the first-order, second-order and third-order neighbors of roads, which are named as MDCGCN-f, MDCGCN-s and MDCGCN-t, respectively, as show in Table 5. The specific equation is as follows:

$$p_{i,j} = \frac{Z\sum x_i^{l-1}x_j^{l-1} - \sum x_i^{l-1}\sum x_j^{l-1}}{\sqrt{Z\sum \left(x_i^{l-1}\right)^2 - \left(\sum x_i^{l-1}\right)^2}\sqrt{Z\sum \left(x_j^{l-1}\right)^2 - \left(\sum x_j^{l-1}\right)^2}} \tag{19}$$

$$A_{i,j}^c = \begin{cases} p_{i,j}, & p_{i,j} \neq 0 \ and \ A_d^1 \neq 0 \ldots \ and \ A_d^C \neq 0 \\ 0, & p_{i,j} = 0 \end{cases} \tag{20}$$

where A_d^C represents all C-order neighbors of the node. $C = \{1,2,3\}$ (Approximately equal to order 0 when C is infinite). When $a_{i,j} \neq 0$, it means that point i can reach point j in the C-order neighbors. $p_{i,j}$ represents the correlation between node i and node j, x_i^{l-1} and x_j^{l-1} represent the time series values of node i and node j in the l-th layer. Each node have the same data volume as Z.

Taking the PEMSD4 dataset as an example, we can see from Fig. 8 that MDCGCN is always better than MDCGCN-f, MDCGCN-s



(a) (b)

Fig. 8. Prediction performance of the model under different traffic pattern relationship graph.

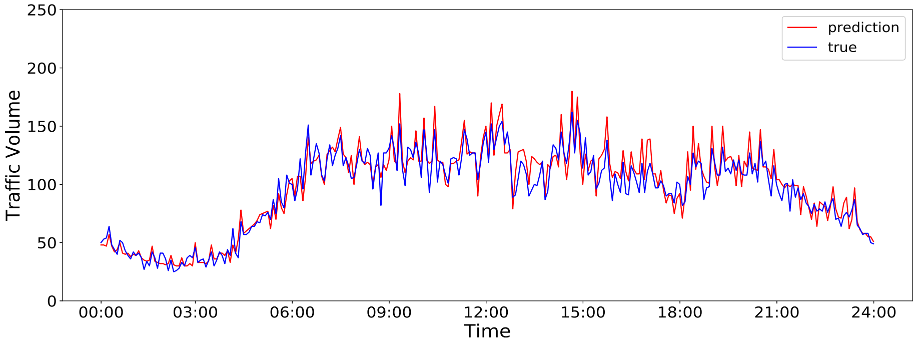


Fig. 9. The visualization results for prediction horizon of 60 min on low traffic volume road.

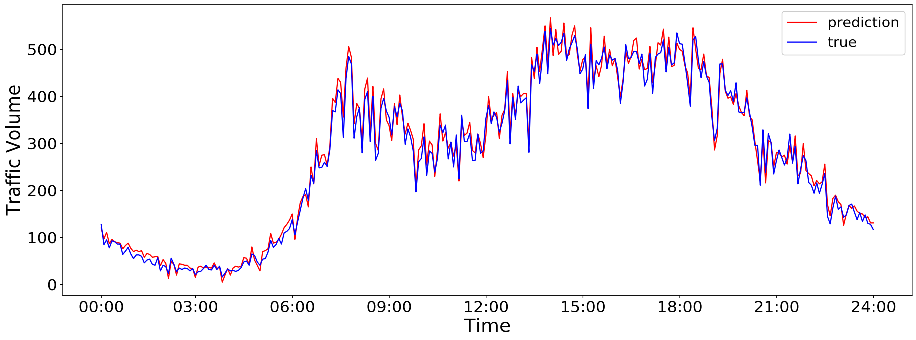


Fig. 10. The visualization results for prediction horizon of 60 min on medium traffic volume road.

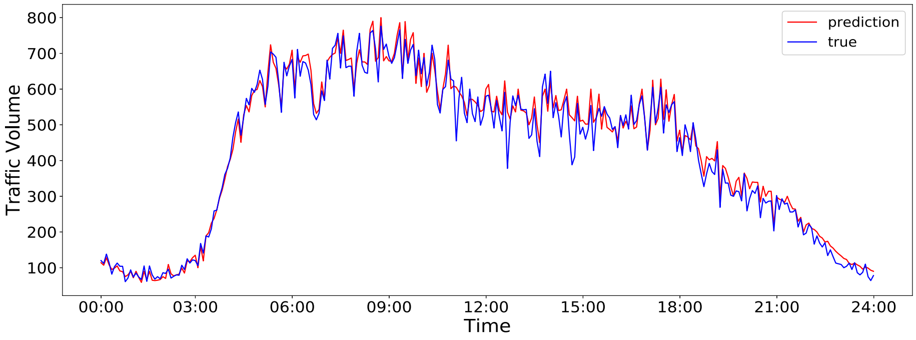


Fig. 11. The visualization results for prediction horizon of 60 min on heavy traffic volume road.

and MDCGCN-t, and when the scope of the constraint gradually expands, the prediction performance gradually approaches the optimal result. Therefore, the experiment shows that road sections with similar traffic patterns, can still have long-term spatio-temporal correlations over time due to the regularity of human activity, even if they are far away in spatial.

5.8. Model interpretation

To better evaluate the practical application prospect of our proposed model, heavy, medium and low traffic loads were selected on PEMSD4 data set and the 60 min predicted results of the test set were compared with the real values, as show in Fig. 9–11. The results show that our model can well predict the peak and valley values of traffic flows and the transformation trend between them, no matter in peak or off-peak periods. However, the overall value will be a little higher, because each node in our figure takes a large amount of traffic information into account. In fact, we believe that high predicted value within a certain range is beneficial for road managers to better implement traffic planning, so our model has a practical application prospect.

6. Conclusion

In this paper, we propose a MDCGCN model for traffic flow prediction. The model combines a benchmark adaptive mechanism and a multi-sensor data connection convolution block. The former considers the difference between periodic data to effectively improve the input quality of the data, and the latter considers the changing relationship between traffic patterns between roads, combining the traffic topology network and traffic pattern relationship graph to capture long-term spatiotemporal correlation. Furthermore, the standard 2D convolution is applied in the temporal dimension while capturing Dynamic spatio-temporal correlation among roads. By extensive comparison experiments, it is verified that the MDCGCN model is suitable for the long-term prediction of more complex traffic network structures, and is superior to existing models in long-term prediction. In addition, it is also found that the proposed benchmark adaptive mechanism can improve the quality of the input and has great significance for improving the overall performance of the model. In future work, we will consider more influencing factors, such as traffic lights and weather conditions on traffic roads, and perform more complex modeling of space–time transportation networks to further improve the prediction accuracy of our model in the short and long term.

CRediT authorship contribution statement

Wei Li: Resources, Supervision. **Xin Wang:** Conceptualization, Methodology, Investigation, Writing - original draft. **Yiwen Zhang:** Resources, Writing - review & editing, Supervision. **Qilin Wu:** Writing - review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the National Science Foundation of China (No. 61872002) and the Natural Science Foundation of Anhui Province of China (No. 1808085MF197). Yiwen Zhang is the corresponding author of this paper.

References

[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, C. Chen, Data-driven intelligent transportation systems: A survey, IEEE Transactions on Intelligent Transportation Systems 12 (4) (2011) 1624–1639.
[2] M.R. Jabbarpour, H. Zarrabi, R.H. Khokhar, S. Shamshirband, K.-K.R. Choo, Applications of computational intelligence in vehicle traffic congestion problem: a survey, Soft Computing 22 (7) (2018) 2299–2320.
[3] M.M. Hamed, H.R. Al-Masaeid, Z.M.B. Said, Short-term prediction of traffic volume in urban arterials, Journal of Transportation Engineering 121 (3) (1995) 249–254.
[4] M.G. Karlaftis, E.I. Vlahogianni, Statistical methods versus neural networks in transportation research: Differences, similarities and some insights, Transportation Research Part C: Emerging Technologies 19 (3) (2011) 387–399.
[5] A.J. Smola, B. Schölkopf, A tutorial on support vector regression, Statistics and Computing 14 (3) (2004) 199–222.
[6] E. Zivot, J. Wang, Vector autoregressive models for multivariate time series, Modeling Financial Time Series with S-Plus® (2006) 385–429..
[7] M.-L. Huang, Intersection traffic flow forecasting based on *v*)gsvr with a new hybrid evolutionary algorithm, Neurocomputing 147 (2015) 343–349.
[8] B.S. Westgate, D.B. Woodard, D.S. Matteson, S.G. Henderson, et al., Travel time estimation for ambulances using bayesian data augmentation, The Annals of Applied Statistics 7 (2) (2013) 1139–1161.
[9] G.A. Davis, N.L. Nihan, Nonparametric regression and short-term freeway traffic forecasting, Journal of Transportation Engineering 117 (2) (1991) 178–188.
[10] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in neural information processing systems, 2016, pp. 3844–3852..
[11] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[12] X. Xu, C. He, Z. Xu, L. Qi, S. Wan, M.Z.A. Bhuiyan, Joint optimization of offloading utility and privacy for edge computing enabled iot, IEEE Internet of Things Journal (2019).
[13] M. Van Der Voort, M. Dougherty, S. Watson, Combining kohonen maps with arima time series models to forecast traffic flow, Transportation Research Part C: Emerging Technologies 4 (5) (1996) 307–318.
[14] B.M. Williams, L.A. Hoel, Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results, Journal of Transportation Engineering 129 (6) (2003) 664–672.
[15] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, H. Li, T-gcn: A temporal graph convolutional network for traffic prediction, IEEE Transactions on Intelligent Transportation Systems (2019).
[16] Y. Sun, B. Leng, W. Guan, A novel wavelet-svm short-time passenger flow prediction in beijing subway system, Neurocomputing 166 (2015) 109–121.
[17] J. Zhang, J. Cui, H. Zhong, Z. Chen, L. Liu, Pa-crt: Chinese remainder theorem based conditional privacy-preserving authentication scheme in vehicular ad-hoc networks, IEEE Transactions on Dependable and Secure Computing (2019).
[18] W.-C. Hong, Traffic flow forecasting by seasonal svr with chaotic simulated annealing algorithm, Neurocomputing 74 (12–13) (2011) 2096–2107.
[19] H. Chang, Y. Lee, B. Yoon, S. Baek, Dynamic near-term traffic flow prediction: system-oriented approach based on past experiences, IET Intelligent Transport Systems 6 (3) (2012) 292–305.
[20] D. Xia, B. Wang, H. Li, Y. Li, Z. Zhang, A distributed spatial–temporal weighted model on mapreduce for short-term traffic flow forecasting, Neurocomputing 179 (2016) 246–263.
[21] F. Moretti, S. Pizzuti, S. Panzieri, M. Annunziato, Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling, Neurocomputing 167 (2015) 3–7.
[22] X. Xu, X. Zhang, X. Liu, J. Jiang, L. Qi, M. Z. A. Bhuiyan, Adaptive computation offloading with edge for 5g-envisioned internet of connected vehicles, IEEE Transactions on Intelligent Transportation Systems..
[23] X. Xu, X. Liu, Z. Xu, F. Dai, X. Zhang, L. Qi, Trust-oriented iot service placement for smart cities in edge computing, IEEE Internet of Things Journal (2019).
[24] Y. Zhang, C. Yin, Z. Lu, D. Yan, M. Qiu, Q. Tang, Recurrent tensor factorization for time-aware service recommendation, Applied Soft Computing 85 (2019), 105762.
[25] W. Huang, G. Song, H. Hong, K. Xie, Deep architecture for traffic flow prediction: deep belief networks with multitask learning, IEEE Transactions on Intelligent Transportation Systems 15 (5) (2014) 2191–2201.
[26] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang, Traffic flow prediction with big data: a deep learning approach, IEEE Transactions on Intelligent Transportation Systems 16 (2) (2014) 865–873.
[27] A. Koesdwiady, R. Soua, F. Karray, Improving traffic flow prediction with weather information in connected cars: A deep learning approach, IEEE Transactions on Vehicular Technology 65 (12) (2016) 9508–9517.
[28] X. Ma, Z. Tao, Y. Wang, H. Yu, Y. Wang, Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transportation Research Part C: Emerging Technologies 54 (2015) 187–197.
[29] J.M.P. Menezes Jr, G.A. Barreto, Long-term time series prediction with the narx network: An empirical evaluation, Neurocomputing 71 (16–18) (2008) 3335–3343.
[30] B. Yang, S. Sun, J. Li, X. Lin, Y. Tian, Traffic flow prediction using lstm with feature enhancement, Neurocomputing 332 (2019) 320–327.
[31] Y. Tian, K. Zhang, J. Li, X. Lin, B. Yang, Lstm-based traffic flow prediction with missing data, Neurocomputing 318 (2018) 297–305.
[32] Z. Zhao, W. Chen, X. Wu, P.C. Chen, J. Liu, Lstm network: a deep learning approach for short-term traffic forecast, IET Intelligent Transport Systems 11 (2) (2017) 68–75.
[33] Y. Zhang, C. Yin, Q. Wu, Q. He, H. Zhu, Location-aware deep collaborative filtering for service recommendation, IEEE Transactions on Systems, Man, and Cybernetics: Systems (2019).
[34] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, Y. Yang, Covering-based web service quality prediction via neighborhood-aware matrix factorization, IEEE Transactions on Services Computing (2019).
[35] J. Zhang, Y. Zheng, D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017..
[36] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, Q. He, Efficient query of quality correlation for service composition, IEEE Transactions on Services Computing (2018).
[37] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, X. Xu, A qos-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems, World Wide Web (2019) 1–23.
[38] W. Chen, B. Liu, H. Huang, S. Guo, Z. Zheng, When uav swarm meets edge-cloud computing: The qos perspective, IEEE Network 33 (2) (2019) 36–43.
[39] Y. Seo, M. Defferrard, P. Vandergheynst, X. Bresson, Structured sequence modeling with graph convolutional recurrent networks, in: International Conference on Neural Information Processing, Springer, 2018, pp. 362–373..
[40] B. Yu, H. Yin, Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI), 2018.
[41] S. Guo, Y. Lin, N. Feng, C. Song, H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 922–929..
[42] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, P. Vandergheynst, The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains, IEEE Signal Processing Magazine 30 (3) (2013) 83–98.
[43] M. Simonovsky, N. Komodakis, Dynamic edge-conditioned filters in convolutional neural networks on graphs, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3693–3702.
[44] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9 (8) (1997) 1735–1780.
[45] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR.org, 2017, pp. 1243–1252..
[46] R. Fu, Z. Zhang, L. Li, Using lstm and gru neural network methods for traffic flow prediction, in: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), IEEE, 2016, pp. 324–328..

[47] C. Li, Z. Cui, W. Zheng, C. Xu, J. Yang, Spatio-temporal graph convolution for skeleton based action recognition, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018..

[48] Z. Cui, K. Henrickson, R. Ke, Z. Pu, Y. Wang, Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting, IEEE Transactions on Intelligent Transportation Systems (2018).

[49] K. Guo, Y. Hu, Z. Qian, H. Liu, B. Yin, Optimized graph convolution recurrent neural network for traffic prediction, IEEE Transactions on Intelligent Transportation Systems PP (99) (2020) 1–12..

[50] Y. Li, R. Yu, C. Shahabi, Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: International Conference on Learning Representations (ICLR), 2017..

[51] J. Demiar, D. Schuurmans, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (1) (2006) 1–30.

Wei Li received her PhD degree in computer science in 2006 from Anhui University. She is a professor in the School of Computer Science and Technology at Anhui University. Her research interests include software engineering, deep learning, computer recognition technology.

Xin Wang received his bachelor degree in computer science and technology in 2016 and now is a master student in the School of Computer Science and Technology at Anhui University. His research interests include machine learning and deep learning.

Yiwen Zhang received his PhD degree in management science and engineering in 2013 from Hefei University of Technology. He is a professor in the School of Computer Science and Technology at Anhui University. His research interests include neural computing, service computing and machine learning. More details about his research can be found athttps://bigdata.ahu.edu.cn.

Qilin Wu received his PhD degree in computer application technology in 2011 from Hefei University of Technology. He is a professor in the School of Information Engineering at Chaohu University. His research interests include resource allocation and optimization for wireless networks, edge computing and service computing.