Information Sciences 609 (2022) 548–564

Contents lists available at ScienceDirect

Information Sciences

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

Multi-mode dynamic residual graph convolution network for traffic flow prediction

Xiaohui Huang [a], Yuming Ye [a,*], Weihua Ding [a], Xiaofei Yang [b], Liyan Xiong [a]

[a] Department of Information Engineering, East China Jiaotong University, Jiangxi, China
[b] Faculty of Science and Technology, University of Macau, Macau, China

ARTICLE INFO

Article history:
Received 9 December 2021
Received in revised form 29 March 2022
Accepted 2 July 2022
Available online 22 July 2022

Keywords:
Graph convolution network
Multi-mode fusion
Traffic flow prediction
Spatio-temporal data

ABSTRACT

Urban traffic congestion is not only an important cause of traffic accidents, but also a major hinder to urban development. By learning the historical traffic flow data, we can forecast the traffic flow of some regions in the future, which is of great significance to urban road planning, traffic management, traffic control and many more. However, due to the complex topology of traffic network and the diversity of influencing factors to traffic flow, the traffic modes are usually complicated and volatile, which makes traffic flow prediction very difficult. In this paper, we propose a new graph convolution neural network, namely Multi-mode Dynamic Residual Graph Convolution Network (MDRGCN), to capture the dynamic impact of different factors on traffic flow in a road network simultaneously. Firstly, we design a multi-mode dynamic graph convolution module (MDGCN), which is employed to capture the impact of different traffic modes by learning two types of relationship matrices. Then, we design a multi-mode dynamic graph convolution gated recurrent unit (MDGRU) to realize the combination of spatial and temporal dependences. Finally, we use a dynamic residual module (DRM) to integrate the orginal traffic data and the spatio-temporal features extracted by the MDGRU module to forecast the future traffic flow. Experimental reulsts conducted on the NYCTaxi and NYCBike datasets validate that the MDRGCN model performs better than the other eight baselines.

1. Introduction

As an important part of intelligent transportation system, traffic flow prediction is vital to traffic business management, traffic planning and so on. How to use the existing massive traffic data to accurately predict traffic flow has attracted a large number of researchers, because it can greatly improve the efficiency of traffic management. Traffic flow prediction, however, is a difficult work since the traffic flow are affected by many factors, such as the intricate traffic road network, weather, and some emergency events.

At present, the mainstream methods for traffic flow prediction can be summarized into two categories: the grid division-based methods and the graph construction-based methods.

(1) **Grid division-based methods**: The grid division based methods usually partition a city into several identical subregions by longitude and latitude. On this basis, the convolution is used to extract the temporal and spatial dependences of traffic flow on the grid map. For example, Sun et al. [1] proposed a multi-branch method based on deep learning, which

* Corresponding author.
E-mail address: yming.yes@foxmail.com (Y. Ye).

https://doi.org/10.1016/j.ins.2022.07.008
0020-0255/

processes the input traffic flow data into a traffic flow matrix, and each element in the matrix represents the traffic flow of each section. Wang et al.[2] proposed a SeqST-GAN model for multi-step spatio-temporal crowd traffic prediction, which converts the traffic data in continuous time periods into a 3-order tensor.

However, the methods based on grid division may destory the natural physical connection of the traffic road network, and cannot capture the spatial dependence of traffic flow well, resulting in the decrease of prediction accuracy.

(2) **Graph construction-based methods**: The graph construction-based methods usually model a city as a traffic graph with nodes and edges, so as to better express the real physical semantic information in a road network. For example, Peng et al. [3] proposed a spatio-temporal incidence dynamic graph neural network to capture deeper traffic flow features. In order to predict the speed of traffic roads in a city, Yu et al. [4] proposed a data-driven deep graph convolution method. Ye et al.[5] proposed a Coupled layer-wise Graph Convolution model (CCRNN), which utilized a data-driven method to learn the dependences of between the stations from historical data, and used an encoder-decoder architecture to achive the prediction of the multi-step prediction.

However, there are several problems that have not been solved well in most graph convolution-based methods. 1) The previous GCN-based methods use only physical distance or connectivity between stations to construct the relationship matrix, and the potential spatial dependence between stations cannot be fully explored. For example, obstacles between two stations may reduce the traffic flow between them, even if the physical distance between the two station is close. It is unreasonable to use only physical distance to learn the spatial dependence between two stations. 2) The most existing GCNs use only one type of relationship matrix to capture the features of traffic flow, but the flow in the traffic network may be affected by multiple traffic modes simultaneously. It is difficult to capture the impact of different factors to the traffic flow using only one relationship matrix. 3) Most existing methods predicts traffic flow using only the extracted features by different kinds of neural network modules. However, the original historical traffic flow may directly contain much useful information. For example, the flow of last time slice may be very similar to that of time slice and the flow of last Sunday may be close to this Sunday.

In fact, the traffic flow of most stations is the result of many factors acting in concert. As shown in Fig. 1, station 2 and station 4 represent two residential areas; station 1 and station 5 represent the gymnasium and official area, respectively. The traffic flow at station 5 may be affected by two different types of traffic flow concurrently. The regular traffic flow of station 5 may be the commute from the directions of stations 2 and 4 at some time intervals. This type of traffic flow is usually fixed and regular (as shown by the black arrow in the Fig. 1). Another type of traffic flow of station 5 may come from the station 1 since there is a gymnasium next to the station 1 (as shown by the black dashed arrow in Fig. 1), where there are occasional concerts or sports events. Therefore, in order to accurately forecast traffic flow, multi-mode methods can be employed to capture the change of traffic flow caused by different factors. The challenges of traffic flow forecasting can be summarized as:

- Spatial and temporal dependences: The traffic flow of a station in a time interval is usually related to the flow of its pervious time intervals and the flow of other stations. How to obtain the periodic change law of traffic flow and the traffic relationship between a station and other stations is the key to accurately predict traffic flow.

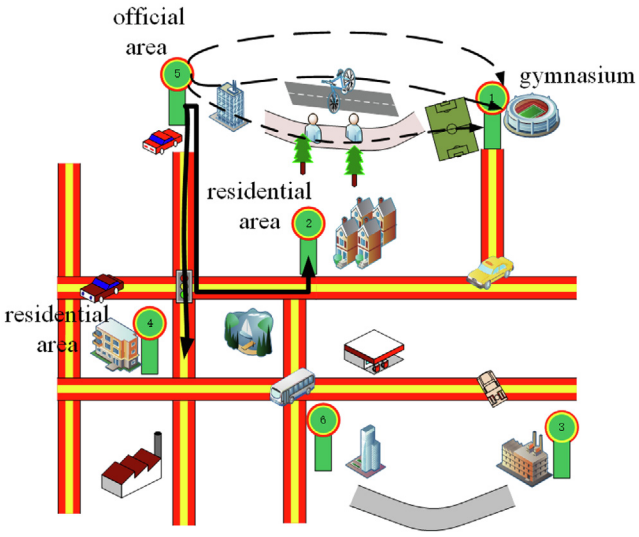


Fig. 1. An example of a traffic network graph. Each station (green circles) in the figure is regarded as the node in the graph.

- Multi-mode relationship extraction and fusion: There may be multiple traffic modes in a traffic network caused by different factors, such as commute, special events, weather etc. How to extract the flow relationships between different stations caused by different factors and fuse the relationships to predict the traffic flow is another challenge.

In order to address the above issues, we propose a Multi-mode Dynamic Residual Graph Convolution Network (MDRGCN) to improve the traffic flow prediction. Firstly, we design a Multi-mode Dynamic Graph Convolution Network (MDGCN), which can learn two kinds of relationship matrices from the historical traffic data rather than only one relationship matrix from physical connection in traditional methods to capture different traffic flow modes, and a dynamically fusing mechanism is presented to merge the features of two modes. Then, we feed the multi-mode features into a dynamic gated recurrent unit (MDGRU) to realize the fusion of temporal and spatial features. Finally, the dynamic residual module (DRM) is used to fuse the historical data and spatio-temporal features to get the final prediction results. In general, our contributions are as follows:

- We propose a novel multi-mode dynamic graph convolution to integrate the flow features of different traffic modes simultaneously.
- We introduce a temporal and spatial fusion module by redesigning the gated recurrent unit (GRU) with multi-mode dynamic graph convolution.
- We propose a dynamic residual fusion module, which can fuse the original information and the decoding information to flow prediction.

The remaining work of this paper is arranged as follows: the related work of traffic flow prediction is placed in Section 2. Section 3 introduces the problem of traffic flow prediction and the definition of graph convolution neural network. The MDRGCN framework and detailed workflow are placed in Section 4. In Section 5, the effectiveness of MDRGCN model is verified by extensive experiments. Finally, the summary of this work and the prospect of future work are put in Section 6.

2. Related work

In this section, we briefly review the previous works related to traffic flow forecasting from the following three aspects: traffic flow prediction based on convolution network (CNN), traffic flow prediction based on recurrent neural network (RNN) and traffic flow prediction based on graph convolution neural network (GCN).

2.1. Traffic flow prediction based on CNN

The CNN-based traffic flow prediction methods usually process the flow data into a grid map, and use the advantages of CNN to mine higher-level traffic flow features in the grid map. In order to capture the complex nonlinear spatial dependence of traffic flow, Yao et al.[6] proposed the Deep Multi-View Spatial–Temporal Network to capture the temporal and spatial dependence of traffic flow at the same time, where the local CNN is used to model the local spatial dependence. Yao et al. [7] proposed a spatio-temporal network with meta-learning to predict urban traffic flow, which utilizes the CNN to extract the spatial dependence between different regions, and uses the LSTM to capture the temporal dependence of each region.

Liang et al. [8] proposed a spatial and temporal relationship network (STRN) to predict fine-grained urban traffic, where CNN is used to learn the high-level representation of each regional grid area, and then a global relationship module (GloNet) is designed to capture the global spatial dependence. Jia et al. [9] proposed a deep spatio-temporal network to predict the traffic flow of urban roads, which uses the dense convolution network to learn the spatial dependences. Compared with the other latest methods, this method achieved higher accuracy in both single-step and multi-step traffic flow prediction tasks.

2.2. Traffic flow prediction based on RNN

As a common method of time series data prediction, Recurrent Neural Network (RNN) is widely used in time series analysis tasks such as machine translation and traffic prediction. In the basis of simple RNN, various improved versions, such as LSTM [10] and GRU [11,12], are proposed to capture longer temporal dependence in data by memory cells. Jiang et al. [13] designed an online system called DeepUrbanEvent, which aims to extract deeper features from the current instantaneous observed traffic flow to generate accurate predictions for the dynamic changes of the population in the short term. The experiment was carried out on multiple large-scale real-life event datasets, and the results confirmed the advantages of DeepUrbanEvent compared to existing methods. Zhu et al. [14] used massive traffic trajectory data to predict the future state of road traffic, where RNN is employed to capture the complex non-linear relationship between traffic roads.

In order to capture the intra-flow dependence and inter-flow correlation of traffic flow, Gao et al. [15] proposed a novel attention-based recurrent neural network model. Gu et al. [16] proposed a fusion deep learning model (FDL) to predict lane-level traffic speed by combining LSTM and GRU to construct a two-layer deep learning framework. In order to capture the spatial dependence between traffic road network, He et al. [17] proposed a spatio-temporal neural network (STNN), which

learns dynamic spatio-temporal dependencies from historical traffic time series through an encoder-decoder structure. The experiment was carried out on real traffic datasets, and the results show that the STNN is better than the advanced models. Although traditional RNN and its improvements can effective obtain periodic patterns hidden in time series data, it is difficult for these types of methods to capture the flow relationship information between different areas or locations in a traffic network.

2.3. Traffic flow prediction based on GCN

As graph has stronger expression ability compared with vector or matrix for non-Euclidean space data, such as social network and knowledge graph, graph neural network is becoming to a type of popular methods to solve various problems of real applications. Graph is also a natural representation of a traffic road network. In order to capture the hidden parttens of traffic flow, Zheng et al. [18] introduced a graph multi-attention network (GMAN), which utilizes the encoder-decoder structure with spatio-temporal attention modules to model the impact of spatio-temporal factors on traffic flow.

Traffic flow is often affected by the context semantics of traffic environment, which increase the difficulty of predicting the overall traffic flow. In order to solve this problem, Dai et al. [19] proposed a hybrid spatio-temporal graph convolution model to improve the spatio-temporal prediction effect by using navigation data. In order to carry out long-short term traffic prediction tasks, Huang et al. [20] proposed a deep learning framework called long-short term graph convolution network. Xie et al. [21] proposed a self-attention based spatio-temporal graph convolution model to capture the highly nonlinear dependence in traffic flow. Chen et al. [22] proposed a traffic prediction framework based on multi-residual recurrent graph neural network (MRes-RGNN) which can extract the temporal and spatial characteristics of traffic flow, so that the model is more sensitive to sudden states. In order to capture the nonlinear interaction between traffic flow and spatial dynamic depedence, Yin et al. [23] proposed a multi-stage attention spatio-temporal graph network. Cui et al. [24] proposed a graph wavelet gated recurrent neural network (GWGRN). In GWGRN, graph wavelet is used to extract spatial features, and the gated recurrent unit is utilized to learn the temporal dependence in traffic series data.

In order to mine the traffic features from fine-grained traffic flow, Fang et al. [25] proposed a fine-grained traffic flow forecasting model based on graph attention network. Xie et al. [26] proposed a deep event perception graph convolution network, which uses the information of traffic events to predict traffic speed. Lu et al. [27] proposed a spatio-temporal adaptive gated graph convolution network (STAG-GCN), which uses the global context information of the road to predict the traffic flow. Bai et al. [28] proposed an adaptive graph convolution recurrent network (AGCRN), which uses adaptive parameter learning module and graph generating module to extract fine-grained spatio-temporal correlation of traffic series data. Liao et al. [29] improved traffic prediction by integrating the offline geographic attributes and road intersection information into the encoder-decoder structure. Zhao et al. [30] proposed a general architecture of spatio-temporal data fusion with parameter efficiency to predict traffic demand, which can obtain more accurate traffic demand prediction results by improving the effectiveness of heterogeneous multi-source data fusion. Peng et al. [31] proposed a graph convolution strategy network of reinforcement learning to learn the dynamic graph structure, so as to capture the spatial characteristics of traffic flow. However, most of existing graph-based methods use only one adjaceney matrix in graph convolution to descripe a type of flow relationship among stations, thus these methods may fail to capture the deeper traffic flow features under the multiple traffic modes.

3. Preliminaries

In this section, we will introduce some notations and concepts of traffic flow forecasting.

3.1. Station-level traffic flow prediction

3.1.1. Representation of a traffic road network

As shown in Fig. 1 in Section 1, the traffic network of a city can be described as a graph $G = (V, E)$, in which the stations are regarded as the nodes V of the graph, the relationships between the traffic flow of two stations are regarded as the edges E connecting the nodes, and each node has a corresponding feature vector which is composed of historical pick-up and drop-off flow of a station. For the connection between two stations, two types of relationship matrices will be defined to capture the different spatial dependencies. Therefore, a general traffic graph is used to describe a city in this work, instead of grids on a map pratitioning by longitude and latitude in traditional CNN or RNN-based methods. By using the advantages of graph convolution to process graph structure data, we can effectively capture the spatial and temporal dependences in the traffic network.

3.1.2. Relationship matrix construction

The relationship matrix is crucial for graph convolution because it determines the aggregating and updating way of the node features. In order to capture the flow features of different traffic modes, we constructed two different relationship matrices, namely, the similarity matrix and the adaptive matrix.

For the construction of the similarity matrix, given the graph signal X of τ time steps, we calculate the similarity of the historical traffic flow among the stations [5] as the weight of the edges in the graph to obtain the similarity matrix A_{sim}:

$$\begin{aligned} &X \overset{similarity}{\rightarrow} \overline{A}_{sim}, \\ &A_{sim} = norm\left(\overline{A}_{sim}\right), \end{aligned} \tag{1}$$

where $\overline{A}_{sim}$ and A_{sim} are similarity matrix and normalized similarity matrix, respectively. The specific process of computing *similarity* is as follows. At first, the original flow data $X \in R^{\tau \times N \times d}$ of τ time steps is converted into two-dimensional matrix $X \in R^{(\tau \cdot N) \times d}$. And then, SVD decomposition is applied to $X \in R^{\tau \times N \times d}$ to obtain two submatrices with the equation:

$$X = X^t \Sigma (X^s)^T \tag{2}$$

where X^t and X^s are two low-rank submatrices, representing time feature and spatial feature of the stations, respectively. In order to reduce the calculation cost, we select only the first 20 rows and 20 columns to represent the top 20 most important features of the stations. And then, we use the Gaussian kernel function to calculate the similarity between stations i and j as the weight of the edge between the two stations:

$$A_{ij}^0 = exp\left(-\frac{\left\|X_i^s - X_j^s\right\|^2}{\varepsilon^2}\right), \tag{3}$$

where ε is the standard deviation of the distance $\left\|X_i^s - X_j^s\right\|^2$. Each element of the A^0 represents the similarity of historical traffic between two stations, which can be used to capture the regular dependency of traffic flow.

For the adaptive matrix, we randomly initialize two learnable node embedding matrices E_1, E_2 [32], and calculate the spatial dependence between the two embedding layers by multiplying E_1 and E_2 to obtain the adaptive matrix A_{ada}:

$$\begin{aligned} &\overline{A}_{ada} = ReLU\left(E_1 E_2^T\right), \\ &A_{ada} = norm\left(\overline{A}_{ada}\right), \end{aligned} \tag{4}$$

where $E_1 \in R^{N \times r}$ and $E_2 \in R^{N \times r}$ are two sub embedding matrices and r is set to 50 in the experiments of this work. The function of *ReLU* is to remove the elements close to 0 in the relationship matrix, so as to reduce the training parameters. We use *norm* [33] operation to normalize A_{sim} and A_{ada}. The specific process of *norm* is:

$$\begin{aligned} &D = \sum_j \overline{A}_{ada}, \\ &D_1 = \tfrac{1}{D}, \\ &\widetilde{A} = D_1 \overline{A}_{ada}. \end{aligned} \tag{5}$$

The *norm* process of A_{sim} is the same as that of A_{ada}.

3.1.3. The definition of traffic flow forecast

Given the input graph signal data X of P time steps, the definition of traffic flow prediction can be described as: through learning the historical traffic graph signal data of P time steps, a mapping function F is obtained to predict the future traffic graph signal data of Q time steps. The formula can be defined as:

$$[X_{t-P+1:t}, G] \overset{F}{\rightarrow} X_{t+1:t+Q}, \tag{6}$$

where $X_{t-P+1:t} \in R^{P \times N \times d}$ and $X_{t+1:t+Q} \in R^{Q \times N \times d}$.

3.2. Graph convolutional network

Graph convolution network is a type of neural network to handle graph structure data. The input of graph convolution network usually includes two parts: the input traffic features $X \in R^{N \times d}$, and the similarity matrix $\widetilde{A}^{sim} \in R^{N \times N}$ and adaptive matrix $\widetilde{A}^{ada} \in R^{N \times N}$ generated by Section 3.1.2 where N represents the number of nodes and d represents the number of features of each node. We first normalize the relationship matrix:

$$\widehat{A} = D^{-1} \widetilde{A} \tag{7}$$

where D is the diagonal matrix of node degrees. According to the literature [34], we establish a K-step diffusion graph convolution on the undirected graph, which can be expressed as:

$$X \overset{\wedge}{\star}_G g_\theta = \sum_{i=0}^{K} \left(\widehat{A}\right)^i X\theta_i, \tag{8}$$

where θ_i represents the parameters of the convolution kernel, K represents the number of diffusion steps in the diffusion graph convolution, and X is the input traffic data.

4. The multi-mode dynamic residual graph convolution network

As shown in Fig. 2,3, the Multi-mode Dynamic Residual Graph Convolution Network contains three modules: Multi-mode Dynamic Graph Convolution Network (MDGCN), Multi-mode Dynamic graph Gated Recurrent Unit (MDGRU) and Dynamic Residual Module (DRM). In the following subsections, we will introduce each module of the MDRGCN model in detail.

4.1. Multi-mode dynamic graph convolution

4.1.1. Coupled graph convolution

Most existing GCN-based methods use fixed relationship matrix in graph convolution. However, there may be different types of traffic modes jointly affecting the change of the traffic flow at the same time, and it is impossible to take into account these influences by using an fixed relationship matrix. To resolve this problem, we employ the Coupled Graph Convolution (CGC) [5] to capture different types of flow patterns, in which different convolution layers have different relationship matrices. The updating process of relationship matrices is defined as:

$$\begin{aligned} Z_{sim}^{(m+1)} &= Z_{sim}^{(m)} *_G g_\theta^{(m)} = \sum_{i=0}^{K} \left(\widetilde{A}_{sim}^{(m)}\right)^i Z_{sim}^{(m)} \theta_i^{(m)}, \\ Z_{ada}^{(m+1)} &= Z_{ada}^{(m)} *_G g_\theta^{(m)} = \sum_{i=0}^{K} \left(\widetilde{A}_{ada}^{(m)}\right)^i Z_{ada}^{(m)} \theta_i^{(m)}, \end{aligned} \tag{9}$$

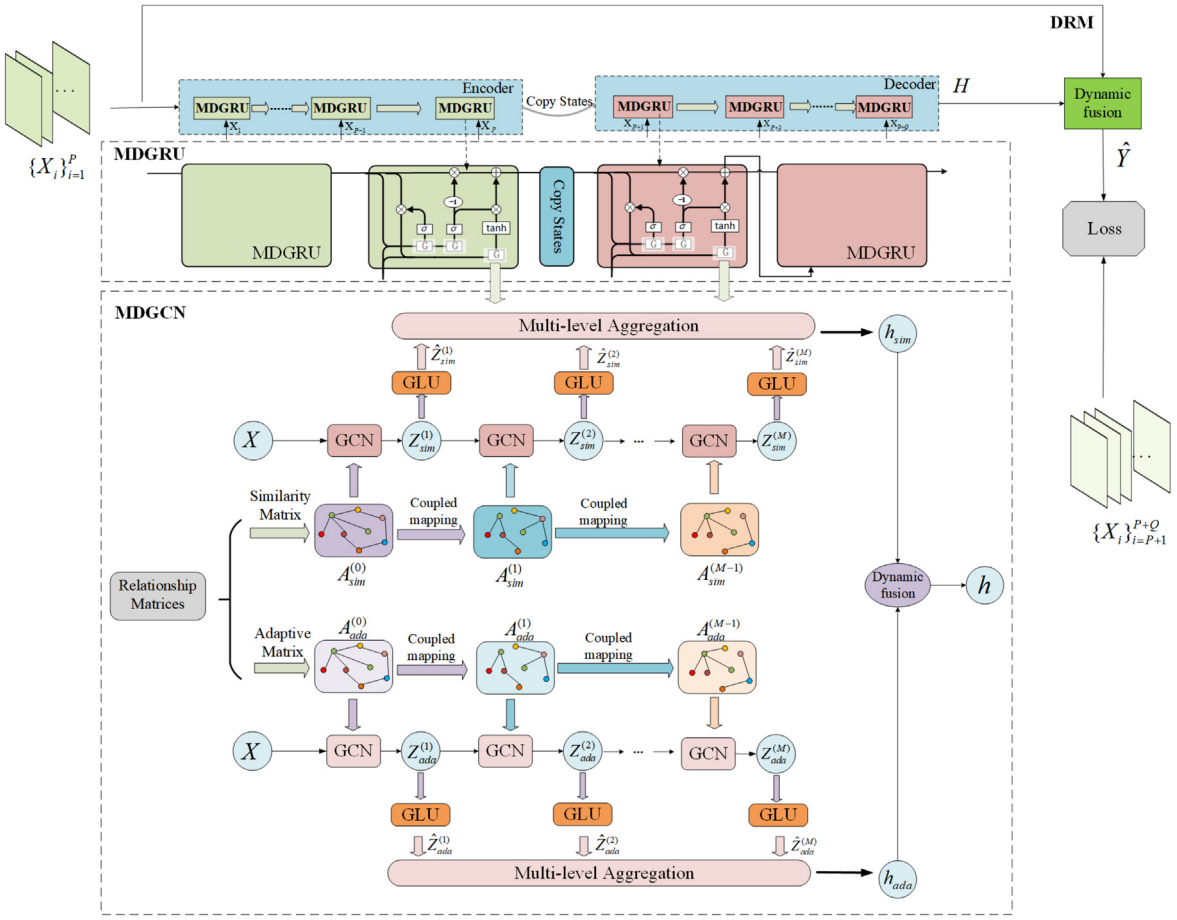


Fig. 2. The framework of the MDRGCN.

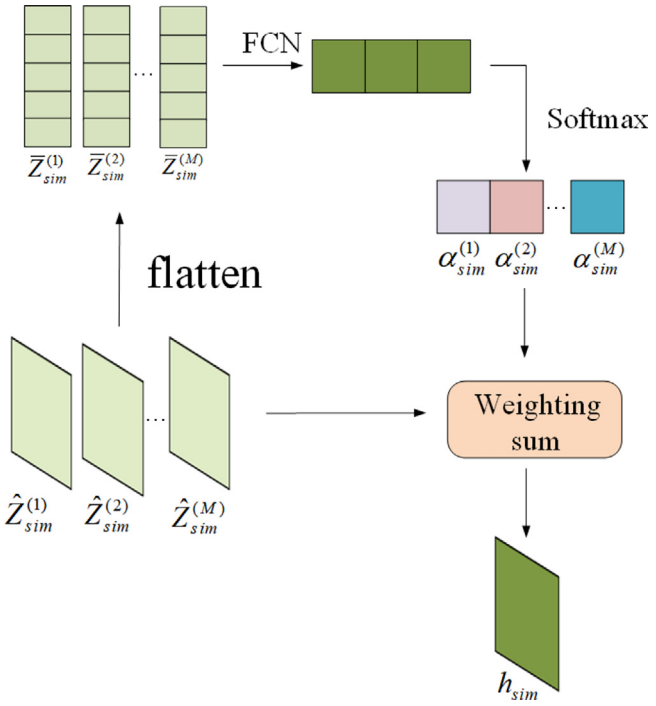


Fig. 3. Multi-level aggragation module.

where $Z_{sim}^{(0)}$ and $Z_{ada}^{(0)}$ are the input data X, and $\theta_i^{(m)}$ represents the parameters of the convolution kernel of the m^{th} layer. A_{sim} can be used to model the fixed and regular flow features between stations, and the adaptive matrix A_{ada} can be used to extract the volatile flow features near the stations. The similarity matrix and adaptive matrix of the $m+1^{th}$ layer can be obtained by updating the matrices of the m^{th} layer as:

$$\begin{aligned} A_{sim}^{(m+1)} &= \psi^{(m)}\left(A_{sim}^{(m)}\right), \\ A_{ada}^{(m+1)} &= \psi^{(m)}\left(A_{ada}^{(m)}\right), \end{aligned} \tag{10}$$

where $A_{sim}^{(m)}$, $A_{ada}^{(m)}$ represents the similarity matrix and adaptive matrix of m^{th} layer. $\psi^{(m)}$ represents the function of updating the matrices of layer m. In the experiment, we use the fully connected network to update the similarity matrix and adaptive matrix, and the specific update process of function $\psi^{(m)}$ is:

$$\begin{aligned} E_1^{(m+1)} &= E_1^{(m)} W^{(m)} + b^{(m)}, \\ E_2^{(m+1)} &= E_2^{(m)} W^{(m)} + b^{(m)}, \\ A^{(m+1)} &= ReLU\left(E_1^{(m+1)}\left(E_2^{(m+1)}\right)^T\right), \end{aligned} \tag{11}$$

where $E_1^{(m)}$ and $E_2^{(m)}$ are the two sub embedding matrices of the m^{th} layer, $A^{(m+1)}$ represents the relationship matrix of the $(m+1)^{th}$ layer, $W^{(m)}$ and $b^{(m)}$ are the weight and bias of the fully connected layer.

4.1.2. Gated graph convolution module

In order to effectively fuse the spatial features extracted by different diffusion steps in each graph convolution, we design a dynamic graph gated mechanism based on the Gated Linear Unit (GLU). The specific process of the dynamic gated linear graph convolution module is:

$$\begin{aligned} Z_{sim,1}^{(m)}, Z_{sim,2}^{(m)} &= split\left(Z_{sim}^{(m)}\right), \\ \widehat{Z}_{sim}^{(m)} &= Z_{sim,1}^{(m)} * sigmoid\left(Z_{sim,2}^{(m)}\right) \end{aligned} \tag{12}$$

where $\widehat{Z}_{sim}^{(m)} \in R^{N\times L}(1 < m < M)$ is the represetations of the stations through the gated linear unit, L represents the feature number of the station. Operation *split* represents that the similarity matrix $Z_{sim}^{(m)} \in R^{N\times 2L}$ is evently divided into equal size

$Z_{sim,1}^{(m)} \in R^{N\times L}$ and $Z_{sim,2}^{(m)} \in R^{N\times L}$ along with the feature dimension. The updating process of adaptive matrix is consistent with that of similarity matrix, and the final output is $\widehat{Z}_{ada}^{(m)} \in R^{N\times L}$.

4.1.3. Multi-level aggregation

In order to collect information from multiple graph convolutional layers, we utilize the attention mechanism to aggregate the information of all layers, so as to select more important information for prediction tasks. After the gated graph convolution unit, the graph signal data can be expressed as $\widehat{Z}_{sim} = \left\{\widehat{Z}_{sim}^{(1)}, \widehat{Z}_{sim}^{(2)}, \ldots \widehat{Z}_{sim}^{(M)}\right\} \in R^{M\times N\times L}$ and $\widehat{Z}_{ada} = \left\{\widehat{Z}_{ada}^{(1)}, \widehat{Z}_{ada}^{(2)}, \ldots \widehat{Z}_{ada}^{(M)}\right\} \in R^{M\times N\times L}$, where M represents the number of layers of the coupled graph convolution, and L represents the dimension of the station feature. The attention score is calculated by a linear function as:

$$\alpha_{sim}^{(m)} = \frac{exp\left(\overline{Z}_{sim}^{(m)} W_\alpha + b_\alpha\right)}{\sum_{m=1}^{M} exp\left(\overline{Z}_{sim}^{(m)} W_\alpha + b_\alpha\right)}, \tag{13}$$

where W_α and b_α are the learnable weights and bias of the linear function, $\overline{Z}_{sim}^{(m)}$ is the one-dimensional flatten of $\widehat{Z}_{sim}^{(m)}$. And then, the operation of multi-level aggregation is formulated as:

$$h_{sim} = \sum_{m=1}^{M} \alpha_{sim}^{(m)} \widehat{Z}_{sim}^{(m)}. \tag{14}$$

where $h_{sim} \in R^{N\times L}$ is the fusion presentation of M layers' features. The adaptive matrix graph convolution aggregation process is consistent with that of the similarity matrix, and the aggregation presentation is denoted as $h_{ada} \in R^{N\times L}$.

4.1.4. Dynamic Fusion Module

After multi-level aggregation, the traffic features of two different modes are extracted. Then we design a dynamic fusion module to fuse the features of two modes, so as to extract the deeper spatial dependences for the flow prediction. The specific process of dynamic fusion is:

$$h = W_{sim} \times h_{sim} + W_{ada} \times h_{ada}, \tag{15}$$

where the weight matrices W_{sim} and W_{ada} are learnable parameters, and h_{sim} and h_{ada} are the flow features of two different modes produced by multi-level aggregation module. The integrating features $h \in R^{N\times L}$ will be used as the input of the multi-mode dynamic graph convolution gated recurrent unit (MDGRU) to fuse the temporal features.

4.2. Dynamic graph convolution Gated Recurrent Unit

As a variant of recurrent neural network, GRU can better handle time series data since it can overcome the problem of gradient vanish to some extent. Inspired by the literature [34], we propose a multi-mode dynamic graph convolution module which is used to replace the linear transformation in GRU for fusing spatial and temporal information in taffic flow data. The dynamic graph convolutional gated recurrent unit (MDGRU) is defined as:

$$\begin{aligned}
r^{(t)} &= \sigma\left(\Theta_r \star_G \left[h^{(t)}, H^{(t-1)}\right] + b_r\right),\\
u^{(t)} &= \sigma\left(\Theta_u \star_G \left[h^{(t)}, H^{(t-1)}\right] + b_u\right),\\
c^{(t)} &= tanh\left(\Theta_c \star_G \left[h^{(t)}, \left(r^{(t)} \odot H^{(t-1)}\right)\right] + b_c\right),\\
H^{(t)} &= u^{(t)} \odot H^{(t-1)} + \left(1 - u^{(t)}\right) \odot c^{(t)},
\end{aligned} \tag{16}$$

where $h^{(t)}$ and $H^{(t)}$ denote the output of the dynamic fusion module and the output of the MDGRU at time step t, respectively, $\odot$ represent the Hadamard product, and σ is the activation function. The reset gate $r^{(t)}$ helps to forget unnecessary information. The update gate $u^{(t)}$ is used to control the output of the MDGRU. $\Theta_r, \Theta_u, \Theta_c$ are the corresponding learnable parameters. In the final multi-step prediction stage, we use the encoder-decoder architecture to capture the global temporal dependence of the traffic flow.

4.3. Dynamic residual fusion mechanism

Most existing methods predicts traffic flow using only the extracted features by different kinds of neural network modules. However, the original historical traffic flow may directly contain much useful information. Based on this, we design a dynamic residual fusion module to fuse the input data with the extracted spatio-temporal features to predict traffic flow. The specific formula of dynamic residual fusion mechanism can be expressed as:

$$\widehat{Y} = H * W_{hid} + X * W_{in} \tag{17}$$

where H represents the final spatio-temporal fusing features produced by the decoder, and X represents the original input data. W_{hid} and W_{in} are learnable parameters.

4.4. Loss function

In our proposed method, we use the root mean square error (*RMSE*) as the loss function of the model:

$$RMSE = \sqrt{\frac{1}{Q}\sum_{i}\left(Y_i - \widehat{Y}_i\right)^2} \tag{18}$$

where Y_i and $\widehat{Y}_i$ represent the ground truth and predicted value, respectively, and Q is the number of time steps of the traffic flow that we plan to predict.

5. Experiment

In order to verify the effectiveness of the MDRGCN model, we conducted experiments on two real datasets on the NYC opendata. Next, we will describe the details of the experiments from following parts: datasets, baselines, evaluation index, comparison of experimental results, ablation experiment and so on.

5.1. Datasets

The datasets used in the experiments are the NYCBike and NYCTaxi datasets, which are shown in Table 1.

NYCTaxi: This dataset includes about 35 million taxi transaction records in New York City. The time span is from April 1, 2016 to June 30, 2016, which includes boarding and alighting time, boarding and alighting location information, travelling distance and other attribute information. In the experiments, we took the data from April 1, 2016 to June 2, 2016 as the training set, the data from June 3, 2016 to June 16, 2016 as the validation set, and the remaining data as the testing set.

NYCBike: This dataset contains the sharing bicycle order records in New York City. The transaction time of the records is from April 1, 2016 to June 30, 2016. The dataset contains the following information: bicycle pick-up point and drop off point, pick up time, drop off time and travelling duration, etc. The division of dataset is consistent with NYCTaxi.

5.2. Data set preprocessing

Before the experiments, the datasets were preprocessed into graph data suitable for model input. The NYCBike dataset is station-based, and each bicycle parking spot can be regarded as a station. The NYCTaxi dataset is generated by a station-free system. Therefore, mining potential sites can help to capture the characteristics of traffic flow. In this work, we used the density peak clustering (DPC) [35] algorithm to find the virtual stations on the NYCTaxi dataset. Each time interval on the two datasets are 0.5 h. Before training, the traffic data are standardized by Z-score standardization. The feature dimension D of each station is 2, which represents the number of pick-up and drop-off, respectively. The historical time steps and the predicted time steps are both 12.

5.3. Baselines

In order to verify the performance of MDRGCN, we have chosen the following eight traffic flow prediction algorithms as comparison.

Table 1
The details of two datasets.

Dataset	NYCTaxi	NYCBike
Start time	4/1/2016	4/1/2016
End time	6/30/2016	6/30/2016
Training set (Days)	63	63
Validating set (Days)	14	14
Testing set (Days)	14	14
Time interval (mins)	30	30
Research area (km)	8.42×14.45	8.42×14.45
Stations	266	250

- HA: The historical average (HA) predicts the future flow by calculating the average flow in a historical time period. For example, to predict the traffic flow from 7:00 to 8:00 in a certain area on Saturday nights, the historical real data from 7:00 to 8:00 on all past Saturday nights in the area are utilized to calculate the average flow.
- XGBoost [36]: Extreme gradient boosting (XGBoost) predicts the traffic flow by establishing several sets of regression tree models.
- FC-LSTM [37]: The long short-term memory network (FC-LSTM) with fully connected hidden units is a well-known network structure that can capture the time dependence of traffic flow data.
- DCRNN [34]: As a classical spatio-temporal graph convolutoin network, DCRNN uses graph convolutional networks to capture spatial dependence, and then uses LSTM to temporal dependence.
- STGCN [38]: STGCN combines graph convolution with 1-D convolution to capture the spatial dependence of traffic flow.
- STG2Seq [39]: STG2seq captures long-short term spatio-temporal dependence by establishing spatio-temporal diagram sequence data.
- GraphWaveNet [32]: GraphWave Net is also a model based on GNN and CNN, which combines diffusion graph convolution with dilated convolution.
- CCRNN [5]: CCRNN proposes a coupling mechanism to update the relationship matrix of different layers, and uses an end-to-end architecture to achieve multi-step traffic flow prediction.

5.4. Evaluation metric

In order to evaluate the final prediction results of the model, we used root mean square error (RMSE), mean absolute error (MAE) and pearson correlation coefficient (PCC) as the evaluation criteria. Their calculation formulas are:

$$\begin{aligned} RMSE &= \sqrt{\frac{1}{n}\sum_{i}(y_i - \hat{y}_i)^2}, \\ MAE &= \frac{1}{n}\sum_{i=1}|\hat{y}_i - y_i|, \\ PCC_{X,Y} &= \frac{E[(X-\mu(X))(Y-\mu(Y))]}{\sigma(X)\sigma(Y)}, \end{aligned} \tag{19}$$

where y_i and $\hat{y}_i$ represent the ground truth and predicted traffic value by the models, respectively. n represents the number of the predicted values. $\mu(X)$ and $\mu(Y)$ represent the mean value of the ground truth and the predicted traffic flow, $\sigma(X)$ and $\sigma(Y)$ respectively represent the standard deviation between the ground truth and the predicted traffic flow.

5.5. Parameter setting

From the Fig. 2, we can see that the model mainly contains three modules: multi-mode dynamic graph convolution, multi-mode dynamic graph convolution GRU, and dynamic residual module. In the experiments, the parameter setting of each module is shown in the Table 2. First, the number of graph convolution layers in the coupled graph convolution module is set to 6, and the number of diffusion steps K of the graph convolution of each layer is 3. For the initial relationship matrices, the dimensions of node embedding are set to 70 and 50 on the NYCTaxi and NYCBike datasets, respectively. Then, we use three layers diffusion graph convolution on MDGCN module. There are 12 identical MDRGU blocks in the encoder to extract the temproral-spatial features of the histrical data, and another 12 identical MDGRU blocks are used to predict the traffic flow of the future 12 time steps. The optimization algorithm used in the training process is *Adam*, and the learning rates on NYCTaxi and NYCBike datasets are set to 0.0015 and 0.0005, respectively.

Table 2
The details of two datasets.

Parameter	Value
MCGC layers	6
Aggregation layer	2
Diffusion steps	3
Dimension of node embedding(NYCTaxi)	70
Dimension of node embedding(NYCBike)	50
MDGRU Layer	1
Batch Size	32
Epoch	200
Optimizer	Adam
Early stop	20
Learning rate (NYCTaxi)	0.0015
Learning rate (NYCBike)	0.0005

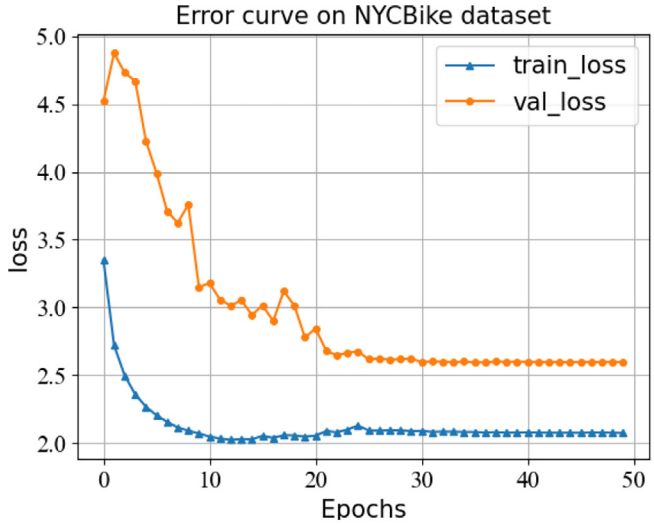
5.6. Convergence analysis

In order to explore the convergence of MDRGCN, we show the error curve during the training and validating process as shown in Fig. 4.

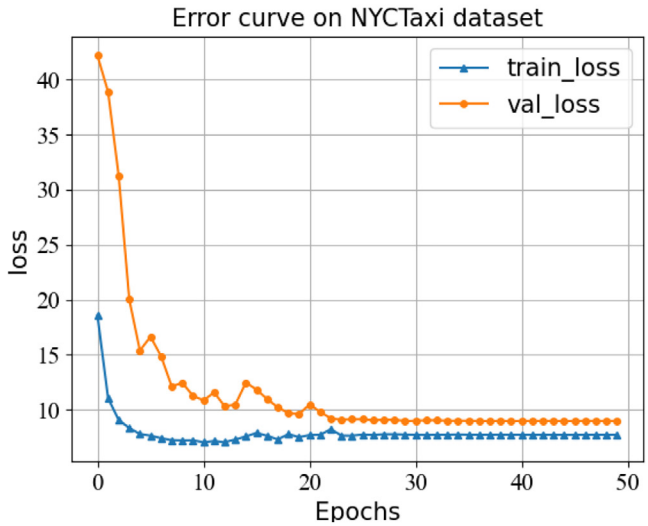
From the Fig. 4, we can see that as the number of training epochs increases, the error continuously decreases, and the model finally converges after 30 epoches on both datasets. In the training process, in order to prevent over-fitting, we use the early stopping strategy, which means that the algorithm stops training when the error of the model on the validating set did not decline.

5.7. Experimental results

Table 3 shows the experimental results of different baselines on the NYCTaxi and NYCBike datasets. From the table, we can see that on the NYCTaxi dataset, the RMSE, MAE, and PCC of the MDRGCN model are imporved 8.17%, 7.67%, and 0.59%,



(a)



(b)

Fig. 4. The training and validation error curves of the MDRGCN model on two data sets: (a) the NYCBike dataset; (b) the NYCTaxi dataset.

Table 3
The comparative results of different models on two datasets.

Method	NYCBike			NYCTaxi		
	RMSE	MAE	PCC	RMSE	MAE	PCC
HA	5.2003	3.4617	0.1669	29.7806	16.1509	0.6339
XGBoost	4.0494	2.4690	0.4861	21.1994	11.6806	0.8077
FC-LSTM	3.8139	2.3026	0.5675	18.0708	10.2200	0.8645
DCRNN	3.2094	1.8954	0.7227	14.7626	8.4274	0.9122
STGCN	3.6042	2.7605	0.7316	22.6489	18.4551	0.9156
STG2Seq	3.9843	2.4976	0.5152	18.0450	9.9415	0.8650
Graph WaveNet	3.2943	1.9911	0.7003	13.0729	8.1037	0.9322
CCRNN	2.8383	1.7404	0.7934	9.5631	5.4979	0.9648
MDRGCN(ours)	**2.7393**	**1.6857**	**0.8077**	**8.7813**	**5.0758**	**0.9705**
imporved	**3.48%**	**3.14%**	**1.80%**	**8.17%**	**7.67%**	**0.59%**

respectively, compared with the best baseline model CCRNN. The models HA, XGBoost and FC-LSTM perform worse than other algorithms since those algorithms only considered the temporal dependence of traffic flow and overlook the relationship informtation of traffic flow among different stations. On the NYCTaxi dataset, although STGCN performs slightly better than XGBoost on RMSE, but the PCC of STGCN are significantly higher than that of XGBoost, which may be reason that STGCN can independently learn the correlation between temporal and spatial features.

Both Graph WaveNet and MDRGCN use an adaptive matrix to capture the flow relationship among different stations, but Graph WaveNet uses only one type of relationship matrix to obtain a type of traffic mode and fails to extract traffic flow features from multiple transportation modes. Similarly, CCRNN only employs the similarity of historical flow between stations to model the spatial dependence without considering the dynamic spatial dependence of traffic flow which maybe cause that the results produced by CCRNN is not as good as MDRGCN.

5.8. Ablation

We also conducted ablation experiments on the MDRGCN by removing or changing some components to verify the effectiveness of each module. Specifically, there are seven variants of MDRGCN:

- W_Skip: Removing the dynamic skip residual module between the input data and the output of the decoder in the MDRGCN model, and only using the decoder features to predict traffic flow.
- W_AdaA: The adaptive matrix in the multi-mode dynamic graph convolution is removed, and only similarity matrix is used to update the graph convolution.
- W_SimA: The similarity matrix in the multi-mode dynamic graph convolution is removed, and only the adaptive matrix is used to update the graph convolution.
- W_GLU: The GLU gating mechanism in multi-mode dynamic graph convolution is removed.
- W_Norm: Removing the norm normalization in the generation of similarity matrix and adaptive matrix.
- GTU2GLU: In the multi-mode dynamic graph convolution, GTU gating unit mechanism [40] is used to filter out unuseful information instead of GLU gating mechanism.
- Sum2DF: In the dynamic graph convolution fusion module, simple sum is used to fuse the input data and the final spatio-temporal features produced by the decoder instead of dynamic fusion.
- GRU2MDGRU: The MDGRU module in MDRGCN is replaced by the traditional GRU, i.e., the linear transformation is used to replace the multi-mode dynamic graph convolution in the MDGRU module.

We can see from the Fig. 5 that on the two datasets, the RMSE and MAE of the MDRGCN model are lower than other variants, which shows the effectiveness of each module of the MDRGCN model to some extent. To further evaluate the predictive ability of the MDRGCN, we show the errors of MDRGCN and its variants at four specific prediction time steps as Table 4. We can observe from the table that the final experimental error produced by the MDRGCN model is lower than those of the seven variants in most of cases, which verifies the validity of the MDRGCN.

Combining with the Fig. 5 and Table 4, we can see that on the dataset NYCTaxi, the MAE and RMSE of MDRGCN are 6.51% and 6.24% lower than sum2DF model, respectively. On the NYCBike dataset, MAE and RMSE of MDRGCN model decreased by 1.44% and 2.5%, respectively, compared with sum2DF model. This reason may be that the dynamical fusion module of MDGCN can better capture the features of traffic flow compared to simple sum. On the dataset NYCTaxi, the MAE and RMSE of MDRGCN decreased by 5.85% and 6.28%, respectively, compared with W_AdaA model. Comparing with W_SimA model, the MAE and RMSE of MDRGCN model decreased by 5.53% and 6.55%, respectively. Removing either of the two relationship matrices will lead to the decline of the final prediction accuracy, which proofs that both of two relationship matrices can play a positive role for traffic flow prediction. On the NYCBike dataset, compared with GRU2MDGRU model, the RMSE and MAE of MDRGCN model decreased by 2.3%. On the data set NYCTaxi, the RMSE and MAE of the MDRGCN model decreased by 4.2%

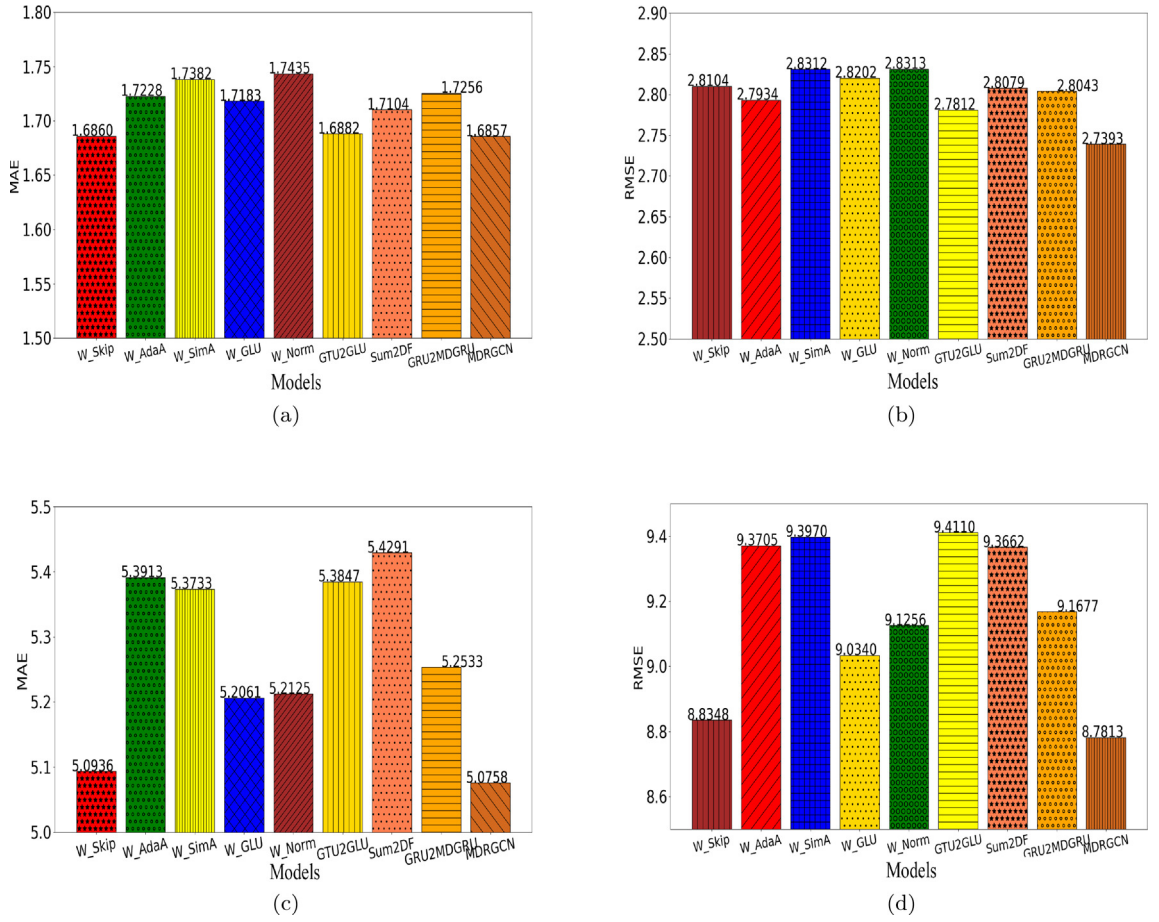


Fig. 5. Experimental results produced by different variants on two datasets. (a) MAE on the NYCBike dataset. (b) RMSE on the NYCBike dataset. (c) MAE on the NYCTaxi dataset. (d) RMSE on the NYCTaxi dataset.

Table 4
Results of ablation experiments at different predicted timestep on two dataset

Method	3^{rd} predicted time step			6^{th} predicted time step			12^{th} predicted time step		
	RMSE	MAE	PCC	RMSE	MAE	PCC	RMSE	MAE	PCC
NYCBike									
W_Skip	2.6237	**1.6013**	0.8275	2.7560	**1.6648**	0.8112	3.0867	1.8148	0.7527
W_AdaA	2.6847	1.6725	0.8156	2.7781	1.7123	0.8068	3.0021	1.8275	0.7685
W_SimA	2.6728	1.6668	0.8173	2.8341	1.7326	0.7978	2.9772	1.8118	0.7699
W_GLU	2.6840	1.6498	0.8228	2.8126	1.7091	0.8090	3.0144	1.8196	0.7642
W_Norm	2.6869	1.6516	0.8238	2.9116	1.7499	0.7967	3.0341	1.8151	0.7702
GTU2GLU	2.6307	1.6113	0.8268	2.7888	1.6849	0.8128	2.9588	1.7890	0.7791
Sum2DF	2.6711	1.6460	0.8213	2.7855	1.6957	0.8139	3.0153	1.8146	0.7722
MDRGCN	**2.6063**	1.6289	**0.8277**	**2.7357**	1.6937	**0.8139**	**2.8953**	**1.7723**	**0.7813**
NYCTaxi									
W_Skip	**8.3999**	**4.9284**	0.9730	9.0232	5.1648	0.9694	9.4515	5.3553	0.9662
W_AdaA	8.8922	5.2366	0.9708	9.3908	5.4206	0.9671	10.3354	5.7616	0.9599
W_SimA	8.869	5.2007	0.9704	9.3543	5.3744	0.9667	10.4457	5.7614	0.9576
W_GLU	8.4870	5.0104	0.9724	9.1748	5.2705	0.9681	9.7634	5.5020	0.9633
W_Norm	8.7767	5.1078	0.9705	9.2222	5.2353	0.9672	9.8294	5.5179	0.9626
GTU2GLU	8.8843	5.1906	0.9702	9.4398	5.4213	0.9663	10.2521	5.7115	0.9593
Sum2DF	9.0885	5.3225	0.9688	9.3829	5.4241	0.9665	10.0874	5.7601	0.9612
MDRGCN	8.4110	4.9742	**0.9730**	**8.7729**	**5.0725**	**0.9706**	**9.4465**	**5.3472**	**0.9663**

and 3.3%, respectively, compared with GRU2MDGRU. The above experimental results validate the effectiveness of MDGCN module. The possible reason of the improvement is that MDGCN module can capture more spatial dependencies among the stations by learning similarity matrix and adaptive matrix than the simple linear transformation in GRU.

5.9. Parameter study

In order to explore the influence of parameters to the performance, we study the dimension of two node embedding matrices E_1 and E_2 of Eq. (4) and the number of hidden units in the MDGRU.

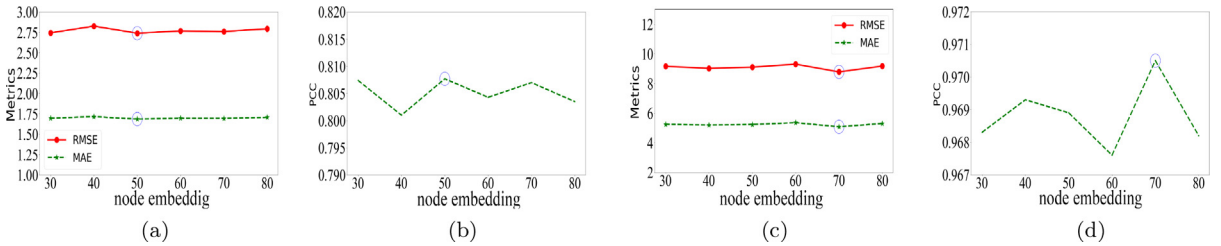


Fig. 6. Comparative results of different node embedding dimensions on two datasets. (a) RMSE and MAE on the NYCBike dataset. (b) PCC on the NYCBike dataset. (c) RMSE and MAE on the NYCTaxi dataset. (d) PCC on the NYCTaxi dataset.

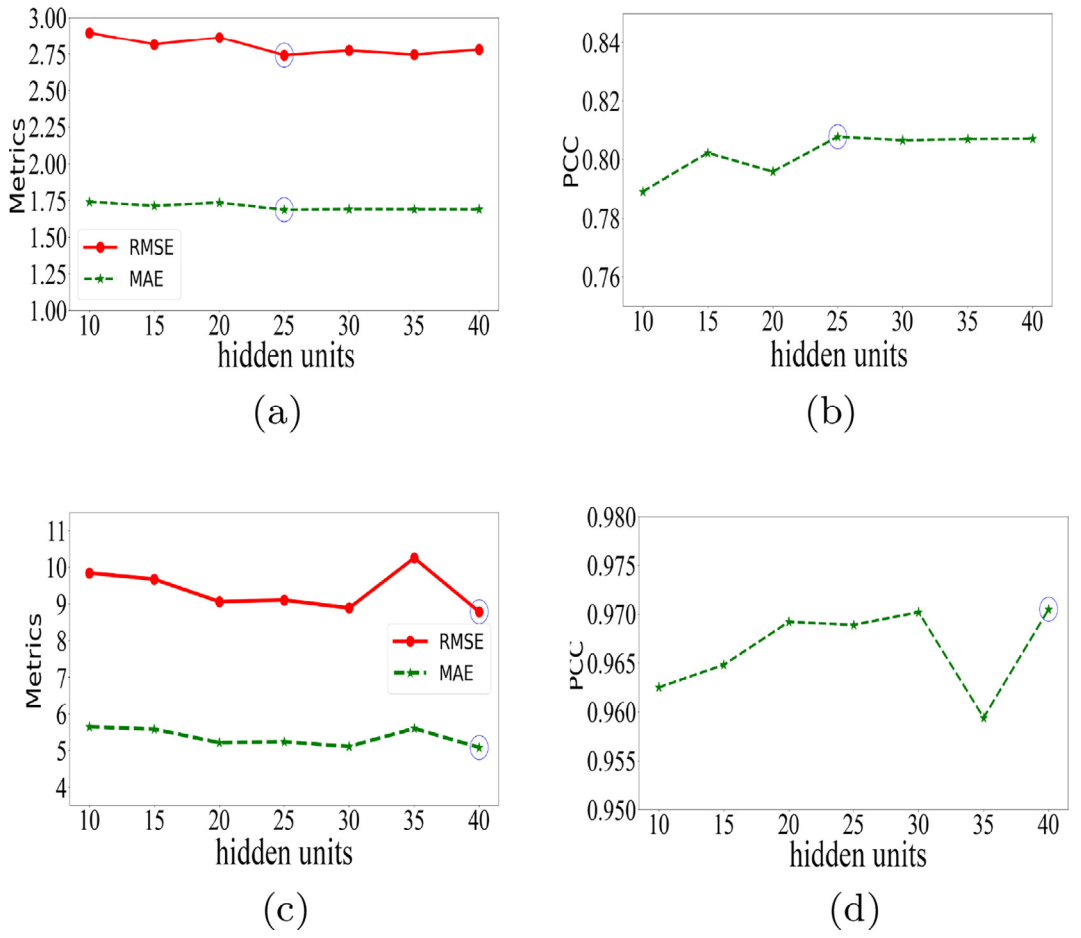


Fig. 7. The results produced by the MDRGCN with different number of MDGRU units on the two datasets. (a) RMSE and MAE on the NYCBike dataset. (b) PCC on the NYCBike dataset. (c) RMSE and MAE on the NYCTaxi dataset. (d) PCC on the NYCTaxi dataset.

5.9.1. Dimension of the node embedding matrices E_1 and E_2

The dimension of the node embedding matrices is an imporant parameter of the proposed algorithm. In order to study the sensibility of the parameter, we conducted experiments with different node embedding dimensions. In the experiments, we choose the dimensions of the node embedding matrices as {30, 40, 50, 60, 70, 80}.

The experimental results corresponding to the different embedding dimensions are shown in Fig. 6. In the figure, the X-axis represents the number of the different node embedded dimensions, and the Y-axis represents the RMSE, MAE or PCC values. From the Figs. 6a and 6b, we can see that when the node embedding dimension is set to 50, the best results of the three evaluation metrics are obtained on the NYCBike dataset. Therefore, the dimension of node embedding on NYCBike dataset is 50 in the comparative experiments. Similarly, as shown in Figs. 6c and 6d, when the node embedding dimension is 70 on the NYCTaxi dataset, the RMSE and MAE results are the lowest and the PCC results are the highest. Therefore, the dimension of node embedding on the NYCTaxi dataset is set to 70 in the comparative experiments.

5.9.2. Numbers of hidden cells in MDGRU module

The number of MDGRU hidden units is a crucial parameter, and different numbers of hidden units may have different effects to the final results. In the experiments, we set the number of hidden units as {10, 15, 20, 25, 30, 35, 40} to study the influence of the parameter to our proposed method.

Fig. 7 shows the error curves of the results produced by the MDRGCN model with the change of the number of hidden units in MDGRU module on the two datasets. As shown in Figs. 7a and 7b, on the NYCBike dataset, when the number of hidden units is set to 25, the RMSE and MAE are the lowest, and the PCC is the highest. Therefore, we finally set the number of MDGRU hidden to 25 on the NYCBike dataset. Similarly, from the Figs. 7c and 7d, we can see that when the number of MDGRU hidden units is set to 40 on the NYCTaxi dataset, the three evaluation metrics can achieve the best results.

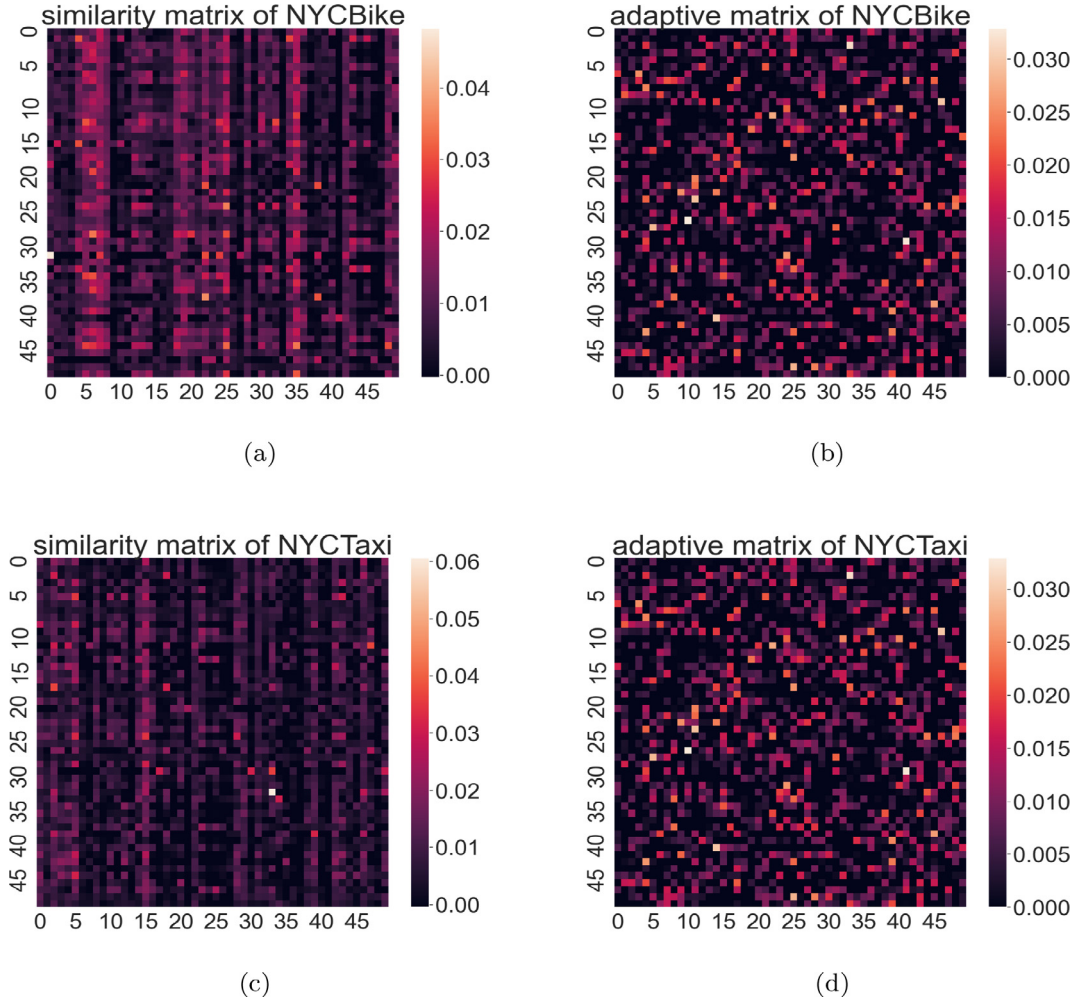


Fig. 8. Heatmaps of two different relationship matrices on the two datasets. (a) Similarity matrix on the NYCBike dataset. (b) Adaptive matrix on the NYCBike dataset. (c) Similarity matrix on the NYCTaxi dataset. (d) Adaptive matrix on the NYCTaxi dataset.

5.10. Traffic mode analysis

In order to further explore traffic modes hidden in flow datasets, we show the two different relationship matrices with heatmap as shown in the Fig. 8. Due to the space limitation, we only show the relationship matrix of the stations 0–49.

As shown in Fig. 8a, the brightly colored columns 6–9, 25 and 35 indicate that the traffic flow of these three stations is highly correlated with the flow of other stations, and there is a large traffic commuting between these stations. The similarity matrix can capture the flow feature with high correlation. From Fig. 8b, we can find that the color distribution of the heat map of the adaptive matrix is scattered. It is worth noting that the $(10, 26), (3, 33)$ and $(29, 41)$ points' colors are the brightest, indicating that the traffic flow correlation between these stations is the highest. Through the combination of the two relationship matrices, we can capture the feature of the regular traffic flow between stations, and also can capture the feature of those randomness between stations. By combining with the flow features of two different traffic modes, we can extract more spatial dependence of traffic flow.

6. Conclusion

This paper proposed a new traffic flow prediction model, named MDRGCN. Specifically, we firstly design a multi-mode dynamic graph convolution module to capture the flow features of different traffic modes through two different relationship matrices, and use the dynamic fusion module to merge the two types of flow features. And then, we embed the multi-mode dynamic graph convolution into the gated recurrent unit to realize the fusion of temporal and spatial dependences. Finally, we fuse the orginal historical traffic data with the final spatio-temporal features produced by the decoder with a dynamic residual connection to obtain the final prediction results. We conducted experiments on NYCTaxi and NYCBike datasets. The experimental results show that the MDRGCN model perform better than other baselines. Further, the ablation experiments verify the effectiveness of each module of MDRGCN.

Although the MDRGCN model can extract and combine the flow features of different traffic modes. However, repeating the same graph convolution process twice is not very efficient and will increase the computation. Therefore, we plan to design an efficient relationship matrix fusion method in the future to reduce the computation while maintaining high prediction accuracy.

CRediT authorship contribution statement

Xiaohui Huang: Conceptualization, Methodology, Writing - original draft. **Yuming Ye:** Data curation, Writing - review & editing. **Weihua Ding:** Software, Validation. **Xiaofei Yang:** Supervision. **Liyan Xiong:** Visualization, Investigation.

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 research was supported by the National Natural Science Foundation of China (Nos.62062033, and 62067002), the Natural Science Foundation of JiangXi Province (No.20212BAB202008).

References

[1] S. Sun, H. Wu, L. Xiang, City-wide traffic flow forecasting using a deep convolutional neural network, Sensors 20 (2) (2020) 1–15.
[2] S. Wang, J. Cao, H. Chen, H. Peng, Z. Huang, SeqST-GAN: Seq2seq generative adversarial nets for multi-step urban crowd flow prediction, ACM Transactions on Spatial Algorithms and Systems 6 (4) (2020) 1–24.
[3] H. Peng, H. Wang, B. Du, H. Ma, J. Liu, L. Wang, Z. Yang, Spatial temporal incidence dynamic graph neural networks for traffic flow forecasting, Information Sciences 521 (2020) 277–290.
[4] Y. James J.Q., "Citywide traffic speed prediction: A geometric deep learning approach," Knowledge-Based Systems, vol. 212, pp. 1–13, 2021.
[5] J. Ye, L. Sun, B. Du, Y. Fu, and H. Xiong, "Coupled layer-wise graph convolution for transportation demand prediction," arXiv preprint arXiv:2012.08080, pp. 1–9, 2020.
[6] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," in Proceedings of the 32nd AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018, pp. 2588–2595.
[7] H. Yao, Y. Liu, Y. Wei, X. Tang, Z. Li, Learning from multiple cities: A meta-learning approach for spatial-temporal prediction, in: Proceedings of the 28th World Wide Web Conference, 2019, pp. 2181–2191.
[8] Y. Liang, K. Ouyang, J. Sun, Y. Wang, J. Zhang, Y. Zheng, D.S. Rosenblum, R. Zimmermann, Fine-grained urban flow prediction, in: Proceedings of the 30th World Wide Web Conference, 2021, pp. 1833–1845.
[9] T. Jia, P. Yan, Predicting citywide road traffic flow using deep spatiotemporal neural networks, IEEE Transactions on Intelligent Transportation Systems 22 (5) (2021) 3101–3111.
[10] X. Ma, Z. Tao, Y. Wang, H. Yu, 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.
[11] D. Chen, H. Wang, M. Zhong, A short-term traffic flow prediction model based on AutoEncoder and GRU, in: Proceedings of the 12th International Conference on Advanced Computational Intelligence, 2020, pp. 550–557.

[12] P. Sun, A. Boukerche, Y. Tao, SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network, Computer Communications 160 (2020) 502–511.
[13] R. Jiang, X. Song, D. Huang, X. Song, Deepurbanevent: A system for predicting citywide crowd dynamics at big events, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019, pp. 2114–2122.
[14] J. Zhu, C. Huang, M. Yang, Context-based prediction for road traffic state using trajectory pattern mining and recurrent convolutional neural networks, Information Sciences 473 (2019) 190–201.
[15] K. Gao, D. Li, L. Chen, J. Geng, F. Gui, Y. Cheng, Y. Gu, Incorporating intra-flow dependencies and inter-flow correlations for traffic matrix prediction, in: Proceedings of the 28th IEEE/ACM International Symposium on Quality of Service, 2020, pp. 1–10.
[16] Y. Gu, W. Lu, L. Qin, M. Li, Z. Shao, Short-term prediction of lane-level traffic speeds: A fusion deep learning model, Transportation Research Part C: Emerging Technologies 106 (2019) 1–16.
[17] Z. He, C. Chow, J. Zhang, STNN: A spatio-temporal neural network for traffic predictions, IEEE Transactions on Intelligent Transportation Systems PP (99) (2020) 1–10.
[18] C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A graph multi-attention network for traffic prediction," in Proceedings of the 34th AAAI Conference on Artificial Intelligence, vol. 34, no. 1, pp. 1234–1241, 2020.
[19] R. Dai, S. Xu, Q. Gu, C. Ji, K. Liu, Hybrid spatio-temporal graph convolutional network: Improving traffic prediction with navigation data, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2020, pp. 3074–3082.
[20] R. Huang, C. Huang, Y. Liu, G. Dai, W. Kong, LSGCN: Long short-term traffic prediction with graph convolutional networks, in: Proceedings of the 29th International Joint Conference on Artificial Intelligence, 2020, pp. 2355–2361.
[21] Y. Xie, Y. Xiong, and Y. Zhu, "SAST-GNN: A self-attention based spatio-temporal graph neural network for traffic prediction," Database Systems for Advanced Applications, pp. 707–714, 2020.
[22] C. Chen, K. Li, S. Teo, X. Zou, and Z. Zeng, "Gated residual recurrent graph neural networks for traffic prediction," in Proceedings of the 33th AAAI Conference on Artificial Intelligence, vol. 33, pp. 485–492, 2019.
[23] X. Yin, G. Wu, J. Wei, Y. Shen, H. Qi, B. Yin, Multi-Stage attention spatial-temporal graph networks for traffic prediction, Neurocomputing 428 (2021) 42–53.
[24] Z. Cui, R. Ke, Z. Pu, X. Ma, Y. Wang, Learning traffic as a graph: A gated graph wavelet recurrent neural network for network-scale traffic prediction, Transportation Research Part C: Emerging Technologies 115 (2020) 1–15.
[25] M. Fang, L. Tang, X. Yang, Y. Chen, C. Li, Q. Li, FTPG: A fine-grained traffic prediction method with graph attention network using big trace data, IEEE Transactions on Intelligent Transportation Systems (2021) 1–13.
[26] Q. Xie, T. Guo, Y. Chen, Y. Xiao, X. Wang, B.Y. Zhao, Deep graph convolutional networks for incident-driven traffic speed prediction, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp. 1665–1674.
[27] B. Lu, X. Gan, H. Jin, L. Fu, H. Zhang, Spatiotemporal adaptive gated graph convolution network for urban traffic flow forecasting, in: Proceedings of the 29th ACM International Conference on Information and Knowledge Management, 2020, pp. 1025–1034.
[28] L. Bai, L. Yao, C. Li, X. Wang, C. Wang, Adaptive graph convolutional recurrent network for traffic forecasting, Advances in Neural Information Processing Systems 33 (2020) 1–16.
[29] B. Liao, J. Zhang, C. Wu, D. McIlwraith, T. Chen, S. Yang, Y. Guo, F. Wu, Deep sequence learning with auxiliary information for traffic prediction, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2018, pp. 537–546.
[30] B. Zhao, X. Gao, J. Liu, J. Zhao, and C. Xu, "Spatiotemporal data fusion in graph convolutional networks for traffic prediction," IEEE Access, vol. 8, pp. 76 632–76 641, 2020.
[31] H. Peng, B. Du, M. Liu, M. Liu, S. Ji, S. Wang, X. Zhang, L. He, Dynamic graph convolutional network for long-term traffic flow prediction with reinforcement learning, Information Sciences 578 (2021) 401–416.
[32] Z. Wu, S. Pan, G. Long, J. Jiang, C. Zhang, Graph wavenet for deep spatial-temporal graph modeling, in: Proceeding of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 1907–1913.
[33] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution networks for traffic forecasting," in Proceedings of the 35th AAAI Conference on Artificial Intelligence, vol. 35, no. 1, 2021, pp. 151–159.
[34] Y. Li, R. Yu, y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in: Proceedings of the 6th International Conference on Learning Representations, 2018, pp. 1–16.
[35] S. Gao, X. Zhou, L.I. Shuai, Clustering by fast search and find of density peaks based on density-raito, Computer Engineering and Applications 208 (2017) 210–217.
[36] T. Chen, C. Guestrin, XGBoost: A scalable tree boosting system, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 785–794.
[37] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[38] 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, 2018, pp. 3634–3640.
[39] L. Bai, L. Yao, S.S. Kanhere, X. Wang, Q.Z. Sheng, STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019, pp. 1981–1987.
[40] Y.N. Dauphin, A. Fan, M. Auli, D. Grangier, Language modeling with gated convolutional networks, in: Proceedings of the 34th International conference on machine learning, 2017, pp. 933–941.