

## Research Article

# A Dynamic Spatio-Temporal Deep Learning Model for Lane-Level Traffic Prediction

Bao Li <sup>1</sup>, Quan Yang <sup>2</sup>, Jianjiang Chen <sup>3</sup>, Dongjin Yu <sup>3</sup>, Dongjing Wang <sup>3</sup>,  
and Feng Wan <sup>3</sup>

<sup>1</sup>Zhejiang Institute of Mechanical and Electrical Engineering Co., Ltd., Hangzhou 311203, China

<sup>2</sup>Zhejiang Testing & Inspection Institute for Mechanical and Electrical Products Quality Co.,Ltd., Hangzhou 310018, China

<sup>3</sup>School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

Correspondence should be addressed to Dongjing Wang; [dongjing.wang@hdu.edu.cn](mailto:dongjing.wang@hdu.edu.cn)

Received 9 November 2022; Revised 1 February 2023; Accepted 24 February 2023; Published 8 March 2023

Academic Editor: Jing Zhao

Copyright © 2023 Bao Li et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Traffic prediction aims to predict the future traffic state by mining features from history traffic information, and it is a crucial component for the intelligent transportation system. However, most existing traffic prediction methods focus on road segment prediction while ignore the fine-grained lane-level traffic prediction. From observations, we found that different lanes on the same road segment have similar but not identical patterns of variation. Lane-level traffic prediction can provide more accurate prediction results for humans or autonomous driving systems to make appropriate and efficient decisions. In traffic prediction, the mining of spatial features is an important step and graph-based methods are effective methods. While most existing graph-based methods construct a static adjacent matrix, these methods are difficult to respond to spatio-temporal changes in time. In this paper, we propose a deep learning model for lane-level traffic prediction. Specifically, we take advantage of the graph convolutional network (GCN) with a data-driven adjacent matrix for spatial feature modeling and treat different lanes of the same road segment as different nodes. The data-driven adjacent matrix consists of the fundamental distance-based adjacent matrix and the dynamic lane correlation matrix. The temporal features are extracted with the gated recurrent unit (GRU). Then, we adaptively fuse spatial and temporal features with the gating mechanism to get the final spatio-temporal features for lane-level traffic prediction. Extensive experiments on a real-world dataset validate the effectiveness of our model.

## 1. Introduction

Intelligent transportation systems (ITS) include driving behaviour understanding [1], path finding [2], map matching [3], and traffic prediction [4]. Traffic prediction refers to predict the future state of traffic by analyzing and mining traffic information in the mining history [5]. As the foundation and important part of ITS, accurate traffic prediction can help formulate real-time control strategies, which is of great importance for scientific planning of traffic management and people's safe and efficient travel [6].

Early efforts in this field use statistical learning methods for traffic prediction, such as differential average moving autoregression [7], which converts unstable sequences into stationary sequences by difference for prediction. Traffic state information has significant nonlinear and uncertain characteristics, and

machine learning methods such as K-nearest neighbor (KNN) [8] and support vector regression (SVR) [9] are also used for traffic prediction. But they have higher requirements for features, which often requires complex feature processing. In recent years, deep learning methods have become the mainstream method of traffic prediction due to automatic feature modeling and effective data mining capabilities. For example, recurrent neural network (RNN)-based methods [10] can effectively model the temporal features in traffic flows. Convolutional neural network (CNN)-based methods [11] regard traffic flows as image and model spatial features or temporal features in European space. The road network or road sensor network is naturally a graph and has a typically non-Euclidean structure. Recently, researchers have used graph-based methods for traffic prediction [12]. CNNs and RNNs can only be used on Euclidean data, while graph-based methods

can effectively model the non-Euclidean structure of graphs for more accurate predictions. With a graph as input, graph-based methods have achieved superior performance in traffic prediction. The topology of the graph is represented by an adjacent matrix, and graph-based methods are directly affected by the adjacent matrix.

Although related work in the past has proposed many effective algorithms in the field of traffic prediction, there are still some limitations and challenges. (1) Previous studies ignore the differences between different lanes and mainly focus on road segment prediction. In reality, there is a wide demand for lane-level traffic prediction. For example, automated vehicles or human-driven vehicles can select appropriate lanes according to the prediction results at the lane level. Traffic congestion can thus be avoided or alleviated [13, 14]. Besides, lane-level traffic prediction can provide more refined and accurate traffic information and help humans or machines make more appropriate and effective decisions. There are different and related patterns of traffic states in different lanes [15–17]. As shown in Figure 1, there are two lanes in both road section 1 and road section 2, and the traffic information between different lanes in the same road section has a similar change pattern, while there still exist some differences in the change pattern in many details. In the road segment-level traffic prediction, the road segment is regarded as a whole, and the prediction results are too macroscopic to provide precise information for lane-level decisions. (2) Graph-based methods rely heavily on adjacent matrices, while most methods build static adjacent matrices, ignoring that the correlation between different nodes on the graph may be different in different situations. For example, there may be similar change patterns for two nodes that are far apart. Besides, the traffic situation of nodes may change in different time periods. It is difficult for the static adjacent matrix to respond timely and effectively to spatio-temporal changes.

To address the aforementioned challenges, we propose a deep learning model for lane-level traffic prediction, which is mainly composed of data-driven GCN and GRU. GCN is used to extract spatial features. To adapt GCN for lane-level traffic prediction, we treat different lanes at the same location as different nodes on the graph. The adjacent matrix of the graph is calculated in a data-driven manner and consists of a traditional distance adjacent matrix and a dynamic lane correlation matrix. GRU is used to extract temporal features. Then, spatio-temporal features are obtained by fusing temporal and spatial features adaptively through the gating mechanism. Finally, lane-level traffic prediction is performed based on the learned spatio-temporal features.

The main contribution of this paper can be summarized as follows:

- (1) A data-driven adjacent matrix is proposed, which consists of a distance-based adjacent matrix and a dynamic lane correlation matrix. It can respond effectively to spatio-temporal changes in a timely manner.
- (2) We propose a deep learning model for lane-level traffic prediction, which learns spatial features with

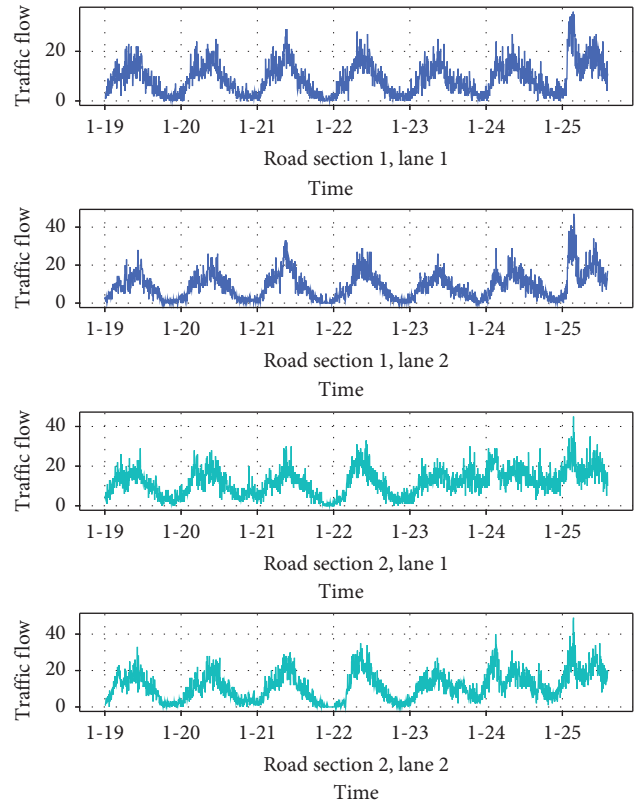


FIGURE 1: An example of lane-level traffic flow, where the horizontal axis represents the traffic flow, and the vertical axis represents the time the traffic flow recorded.

GCN, learns temporal features with GRU, and obtains fused adaptive spatio-temporal features with the gating mechanism.

- (3) Extensive experiments on a real-world dataset validate the effectiveness of the model.

The remainder of this paper is organized as follows: In Section 2, we introduced the related work, which includes general methods and deep learning methods. Section 3 formulates the lane-level traffic prediction task. Section 4 introduces the construction of the data-driven adjacent matrix and the architecture of our model in detail. The comprehensive experiment result on a read-world dataset is demonstrated in Section 5. Finally, we conclude the paper and present future work in Section 6.

## 2. Related Work

**2.1. General Methods.** Traditional traffic prediction methods can be divided into parametric methods and nonparametric methods [10]. Parametric methods rely on the assumption of data stationarity and provide explicit formulations for valuable interpretations of traffic characteristics. Classical parametric methods, such as the autoregressive integrated moving average model (ARIMA) and its variants [7, 18, 19], have been proven to be effective in many scenarios. For example, some studies have found that ARIMA can model highway time series data with high precision [20]. Some

other parametric methods include exponential [21], multivariate time series models [22], and Kalman filtering models [23]. However, the dependency on stationarity makes parametric methods difficult to effectively model the uncertainty and irregular volatility of traffic data. The structure and parameters of the nonparametric methods are not fixed, and the data requirements are not as strict as those of the parameterized methods. The nonparametric methods are more able to deal with complex data such as noisy data and missing data [24]. Typical nonparametric methods include support vector regression [8], K-nearest neighbor [9], the Bayesian network [25], the extreme gradient boost [26], and artificial neural networks (ANN) [24, 27, 28]. Among them, ANN can mine the latent information of traffic data and has nonlinear modeling ability, which is one of the most widely used nonparametric methods. Although nonparametric methods have some achievements in the field of traffic prediction, these methods are limited in their ability to predict lane-level traffic. Besides, both parametric and nonparametric methods are mainly used to model the temporal features and are weak in modeling the spatial features.

**2.2. Deep Learning Methods.** With the rapid development of high-performance data storage and processing technologies, traffic prediction is moving from nonparametric methods to deep learning methods [10]. An important step in traffic prediction is to extract spatio-temporal features from traffic data. For the recurrent neural network (RNN) and its variants like long short-term memory (LSTM) [29] and gated recurrent unit (GRU) [30] which can effectively utilize temporal data, RNN-based methods [31] play an important role in mining temporal traffic features. Ma et al. [32] first applied LSTM to solve the prediction of highway traffic speed and flow. Zhao et al. [10] utilized GRU, which has fewer neurons than LSTM, for traffic prediction. Gu et al. [20] built a fusion system to capture temporal features. RNN-based methods [33, 34] have shown promising results in traffic prediction field, while they are not good at mining spatial features in traffic flow. In terms of spatial traffic features, traffic flows in nearby locations are often strongly correlated [35]. For the power of handling image data, CNN has been used in traffic prediction by treating the traffic flow data as an image. Ke et al. [36] constructed a multichannel CNN model for multilane traffic speed prediction. Liu et al. [37] developed an attention-based CNN structure for traffic speed prediction with the use of traffic flow, speed, and occupancy. However, CNN and RNN can only be applied to Euclidean data; they cannot model the topological structure of the road network or the road sensor network. Neither CNN-based methods nor RNN-based methods are perfect for spatio-temporal feature extraction.

The road network or the road sensor network is naturally a graph. Recently, researchers have applied graph neural networks (GNN), especially graph convolutional networks (GCN) [38], for traffic prediction, and they have superior performance compared to previous approaches. For the ability to model non-Euclidean graph structures, GNNs are

ideal for solving traffic prediction problems. Li et al. [39] treated the traffic flow as a diffusion process and proposed DCRNN, which uses bidirectional random walks on the graph and GRU to capture spatial and temporal features, respectively. Zhao et al. [10] proposed T-GCN, which stacks GCN and GRU for traffic prediction. Yu et al. [40] proposed STGCN to extract spatio-temporal features with complete convolutional structures. Guo et al. [41] established a HGCN model which operates the convolution operation on both micro- and macrotraffic graphs. Zhu et al. [42] employed GCN in multigraph to analyze correlations from multiple perspectives. Guo et al. [43] proposed a dynamic GCN for traffic prediction on the basis of Laplace matrix estimation. Cao et al. [44] combined self-attention with GCN for traffic flow prediction. Although there is a lot of excellent work for traffic prediction, most of them are not suitable for lane-level traffic prediction. Besides, most existing works treat the road or sensor network as a static graph. We propose a deep learning model for lane-level traffic prediction with a dynamic adjacent matrix driven by data. As for lane-level works, Gu et al. [20] combined LSTM and GRU for lane-level traffic speed prediction. Ke et al. [36] introduced a two-stream multichannel CNN model. Ma et al. [45] proposed a convolutional LSTM network for multilane short-term traffic forecasting. Lu et al. [46] described a mix deep learning model for lane-level traffic speed forecasting. Wang et al. [47] presented a heterogeneous graph convolution model for lane-level traffic flow prediction. Existing lane-level traffic prediction methods mostly use RNN or CNN to model spatial features, which has certain limitations.

### 3. Problem Formulation

In this work, we aim to predict the traffic state of lanes in a period of time on the basis of the historical traffic state information recorded on the road sensors. Traffic state is a general concept that includes traffic speed, traffic flow, and other numerical information related to the road. Specially, we predict lane-level traffic flow in the experiment section.

*Definition: Lane Network  $G$ .* To describe the non-Euclidean structure of the lane network, we define it as graph  $G = (\mathbf{V}, \mathbf{E}, \mathbf{A})$ . On graph  $G$ ,  $\mathbf{V} = \{v_1, v_2, \dots, v_N\}$  is the set of nodes, where  $v_i$  represents the  $i$ -th lane and  $N$  is the number of lanes. Note that we treat different lanes on the same road section as different nodes.  $\mathbf{E}$  is the set of edges. The edge between lane  $i$  and lane  $j$  only exists if their distance is less than a certain threshold and there exists a traffic flow from  $v_i$  to  $v_j$ . To better represent the real situation, we consider the traffic flow between different lanes in the same section of the road to be interconnected.  $\mathbf{A} \in \mathbb{R}^{N \times N}$  is the adjacent matrix.

Let  $\mathbf{X}_t = \{x_t^1, x_t^2, \dots, x_t^N\} \in \mathbb{R}^N$  represent the traffic flow of  $N$  lanes on each time stamp  $t$ . Suppose the traffic flow data is the graph signal of  $G$ , given time  $t$  and lane network  $G$ , the lane-level traffic flow prediction problem in our work can be defined as

$$[\mathbf{X}_{t+1}, \dots, \mathbf{X}_{t+T}] = f([\mathbf{X}_{t-p+1}, \dots, \mathbf{X}_t], G), \quad (1)$$

where  $f$  represents the learned mapping function,  $p$  is the input sequence length, and  $T$  is the predicted sequence length.

The key symbols used in this paper are summarized in Table 1.

#### 4. The Proposed Approach

In view of the lack of work on lane-level traffic prediction, this paper proposes a lane-level traffic prediction model. The architecture of our model is illustrated in Figure 2. Specifically, we first establish a data-driven adjacent matrix that can respond to spatio-temporal changes based on the geographic location and historical traffic information of the sensor. The data-driven adjacent matrix is fed into the graph convolutional network (GCN) to capture spatial features, and we model the temporal features with a gated recurrent unit (GRU) model. Then, we adaptively fuse the spatial and temporal features with the gating mechanism to get comprehensive spatio-temporal features. Finally, we make multistep lane-level traffic predictions based on the spatio-temporal features.

**4.1. Data-Driven Adjacent Matrix.** The graph depicts the topological relationship structure between nodes through the adjacent matrix, and the construction of the adjacent matrix directly affects the expressive power of the graph [48]. However, most GCN-based traffic prediction works only construct a static adjacent matrix with fixed weights, without considering that the relationship of different nodes may change in various situations. In particular, it is difficult for a static adjacent matrix to respond to spatio-temporal changes in a timely manner, which makes the model hardly achieve accurate prediction. In our work, we propose a data-driven dynamic adjacent matrix, which is composed of the basic distance-based adjacent matrix  $\mathbf{A}_d$  and the dynamic node correlation matrix  $\mathbf{A}_c$ .

The graphs include directed graph and undirected graph. For undirected graphs such as social networks, the adjacent matrix is symmetric. In the road sensor network, the traffic flows on roads have directions due to the restriction of traffic rules. Graph  $G$  is a directed graph, and the adjacent matrix is asymmetric.

For the basic distance-based adjacent matrix  $\mathbf{A}_d$ , as most works did [49], we calculate one element  $A_{ij}^d$  in  $\mathbf{A}_d$  with

$$A_{ij}^d = \begin{cases} \exp\left(\frac{-d_{ij}^2}{\sigma^2}\right), & \text{if } d_{ij} \leq \varepsilon, \\ 0, & \text{if } d_{ij} > \varepsilon, \end{cases} \quad (2)$$

where  $A_{ij}^d$  represents the influence degree of lane  $v_i$  on lane  $v_j$ ,  $d_{ij}$  is the distance between  $v_i$  and  $v_j$ , and  $\sigma$  is the standard deviation of  $d$ . The distance between different lanes on the same road segment is 0.  $A_{ij}^d$  has a positive value only if  $d_{ij}$  is smaller than threshold  $\varepsilon$  and from  $v_i$  to  $v_j$  exists a traffic flow.

To compensate for the defects caused by the static characteristics of the distance-based adjacent matrix, we

TABLE 1: Summary of notations.

Symbol	Description
$G$	Lane network
$\mathbf{V}, \mathbf{E}$	Node set and edge set of in $G$
$\mathbf{X}_t$	Traffic information of all lanes at timestamp $t$
$x_t^i$	Traffic information of lane $i$ at timestamp $t$
$\hat{\mathbf{A}}$	Data-driven adjacent matrix
$\mathbf{A}_d$	Distance-based adjacent matrix
$\mathbf{A}_c$	Traffic information similarity matrix
$p, T$	Sequence length for input and predict
$\alpha$	A constant that controls the contribution of $\mathbf{A}_c$
$\hat{\mathbf{D}}$	Degree matrix
$\mathbf{u}, \mathbf{r}$	Update gate and reset gate in GRU
$\mathbf{c}, \mathbf{h}$	Cell state and hidden state in GRU
$\mathbf{W}$	Learnable parameter matrices
$\mathbf{g}$	Feature fusion gate
$\mathbf{H}^s$	Learned spatial features
$\mathbf{H}^t$	Learned temporal features
$\mathbf{H}$	Fused spatio-temporal features

further introduce the dynamic correlation matrix  $\mathbf{A}_c \in \mathbb{R}^{N \times N}$ .  $\mathbf{A}_c$  is filled with the Pearson correlation coefficient calculated from the observed input data of the lanes. To be specific,  $\mathbf{A}_c$  at time  $t^*$  is calculated with

$$A_{ij}^c = \frac{\sum_{t=t^*-h+1}^{t^*} (x_t^i - \bar{x}^i)(x_t^j - \bar{x}^j)}{\sqrt{\sum_{t=t^*-h+1}^{t^*} (x_t^i - \bar{x}^i)^2 \sum_{t=t^*-h+1}^{t^*} (x_t^j - \bar{x}^j)^2}}, \quad (3)$$

where  $i$  and  $j$  are the index of lane  $v_i$  and lane  $v_j$ ,  $x_t^i$  is the value of traffic flow on  $v_i$  observed at time  $t$ ,  $\bar{x}^i$  and  $\bar{x}^j$  are means of  $v_i$  and  $v_j$ , respectively. The absolute value of  $A_{ij}^c$  is closer to 1, the higher the correlation between  $v_i$  and  $v_j$ .

Combining the basic distance-based adjacent matrix  $\mathbf{A}_d$  and the dynamic correlation matrix  $\mathbf{A}_c$ , we propose the data-driven adjacent matrix  $\hat{\mathbf{A}}$ ,

$$\hat{\mathbf{A}} = \mathbf{A}_d + \alpha \mathbf{A}_c, \quad (4)$$

where  $\alpha$  is a constant that controls how much  $\mathbf{A}_c$  contributes to  $\hat{\mathbf{A}}$ . On the one hand,  $\hat{\mathbf{A}}$  provides geographic relationships that are fundamental and important for spatial feature extraction; on the other hand,  $\hat{\mathbf{A}}$  can implement timely adjustments to the adjacent matrix with reference to changes of historical information.

**4.2. Spatial Feature Modeling.** Spatial features play an important role in traffic prediction for traffic flow sequences at different locations with connection to some extent. Before the employment of graph-based methods, research studies usually extract the spatial features with multivariate time series models or CNNs [50]. However, limited by the structure, multivariate time series models mostly cannot model the nonlinear relationships between different sequences. Although CNN-based methods can alleviate the situation, the architecture of CNN is bounded to Euclidean space, which is not enough for lane network's topological structure modeling. Recently, graph-based methods have attracted wide attention for their ability in modeling non-

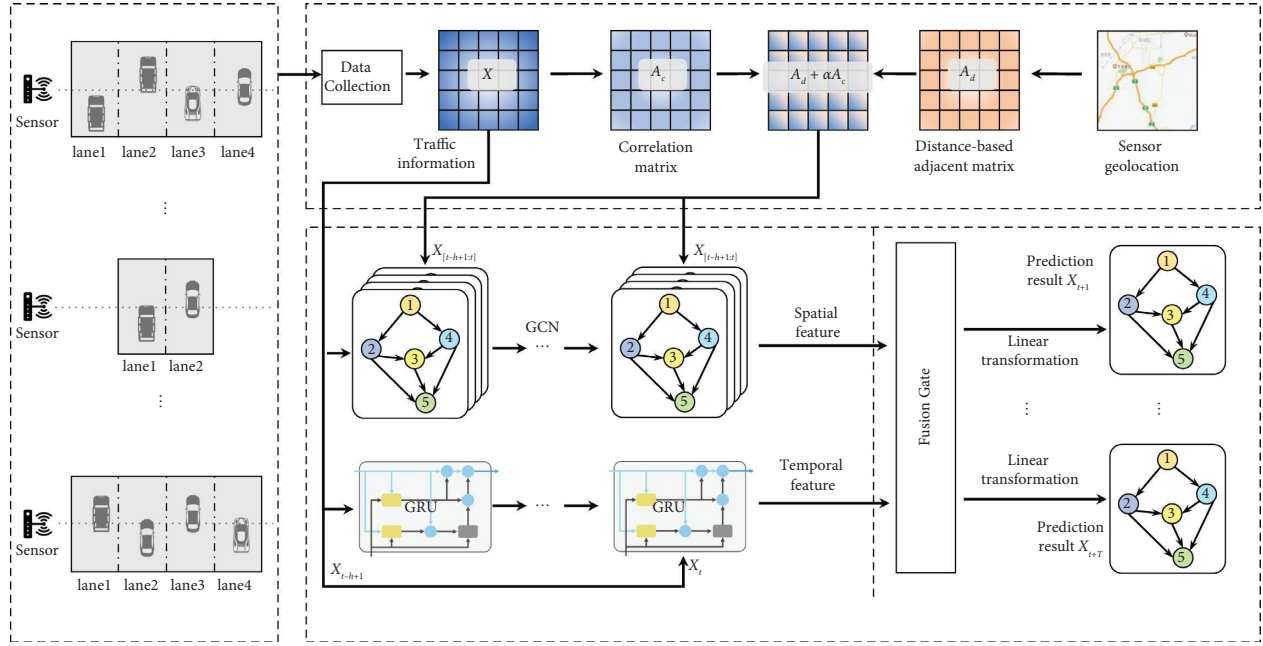


FIGURE 2: The architecture of our proposed model.

Euclidean structure. Specifically, we extract the spatial features with GCN. The GCN model built a filter in the spatial domain, and the spatial features between different nodes on a graph are extracted with the usage of filter. As illustrated in Figure 3, the central node models the topological relationship by aggregating the information of its neighboring nodes in GCN. The topological structure of the graph is encoded to acquire spatial features.

After the construction of the data-driven adjacent matrix  $\hat{\mathbf{A}}$ , the GCN model extracts spatial features with

$$\mathbf{H}_s^{(l)} = \sigma\left(\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2} \mathbf{H}_s^{(l-1)} \mathbf{W}^{(l-1)}\right), \quad (5)$$

where  $\mathbf{H}_s^{(l)}$  represents the feature matrix in  $l$ -th layer,  $\hat{\mathbf{D}}$  is the degree matrix and  $\hat{\mathbf{D}}_{ii} = \sum_j \hat{\mathbf{A}}_{ij}$ .  $\hat{\mathbf{D}}^{-1/2} \hat{\mathbf{A}} \hat{\mathbf{D}}^{-1/2}$  denotes normalize the adjacent matrix  $\hat{\mathbf{A}}$ , which can help to keep the distribution of the feature matrix during the information transfer process.  $\mathbf{W}^l$  is the learned weight matrix in  $l$ -th layer and  $\sigma$  represents the activation function. Note that  $\mathbf{H}^{(0)} = \mathbf{X}$  and  $\mathbf{X}$  is the input traffic information matrix.

**4.3. Temporal Feature Modeling.** Traffic data have significant sequence structure, which means that temporal features are the key to traffic prediction. The recurrent neural network (RNN) is widely used in the processing of sequence data. However, RNN has the problem of gradient disappearance or gradient explosion with the sequence length growing. To solve this problem, researchers proposed many variants of RNN such as LSTM and GRU. These variants have been proved to be effective in sequence data modeling. Both LSTM and GRU control the dissemination and update of information with the gating mechanism. Compared with LSTM, GRU has smaller training time, fewer parameters, and relatively simpler structure. Therefore, we employ GRU

to extract the temporal features of the traffic data. There are two gates in GRU, which are

$$\begin{aligned} \mathbf{u}_t &= \sigma(\mathbf{W}_{ux} \mathbf{x}_t + \mathbf{W}_{uh} \mathbf{h}_{t-1} + \mathbf{b}_u), \\ \mathbf{r}_t &= \sigma(\mathbf{W}_{rx} \mathbf{x}_t + \mathbf{W}_{rh} \mathbf{h}_{t-1} + \mathbf{b}_r), \end{aligned} \quad (6)$$

where  $\mathbf{u}_t$  represents the update gate and controls how much history information the current moment has,  $\mathbf{r}_t$  represents the reset gate and controls how much history information needs to be forgotten.  $\mathbf{x}_t$  is the traffic information at time  $t$ , and  $\mathbf{h}_{t-1}$  is the hidden state at time  $t-1$ .  $\mathbf{W}_{ux}$ ,  $\mathbf{W}_{uh}$ ,  $\mathbf{W}_{rx}$ ,  $\mathbf{W}_{rh}$ ,  $\mathbf{b}_u$ ,  $\mathbf{b}_r$  are learnable parameters. With the gating signals of  $\mathbf{u}_t$  and  $\mathbf{r}_t$ , the cell state  $\mathbf{c}_t$  and output hidden state  $\mathbf{h}_t$  can be calculated with

$$\begin{aligned} \mathbf{c}_t &= \tanh(\mathbf{W}_{cx} \mathbf{x}_t + \mathbf{W}_{ch} (\mathbf{r}_t \odot \mathbf{h}_{t-1})), \\ \mathbf{h}_t &= (1 - \mathbf{u}_t) \odot \mathbf{h}_{t-1} + \mathbf{u}_t \odot \mathbf{c}_t, \end{aligned} \quad (7)$$

where  $\mathbf{c}_t$  can be regarded as the history information stored at time  $t$ , and  $\odot$  represents element-wise multiplication.

As shown in Figure 4, GRU models the hidden state in time  $t$  by taking the current traffic information and the last hidden state  $\mathbf{h}_{t-1}$ . With this operating mechanism, GRU is capable of retaining the historical information while utilizing current traffic information, and then it is able to model temporal features.

**4.4. Our Model.** With GCN and GRU, we obtain spatial and temporal features, respectively. Our next target aims to fuse the spatial and temporal features into comprehensive spatio-temporal features. A convenient operation is to add these two kinds of feature directly, which may weaken the characteristics of the spatial and temporal features themselves. Inspired by the gating mechanism, we introduce a learnable gate  $\mathbf{g}$  to fuse features. Similarly, to the role of

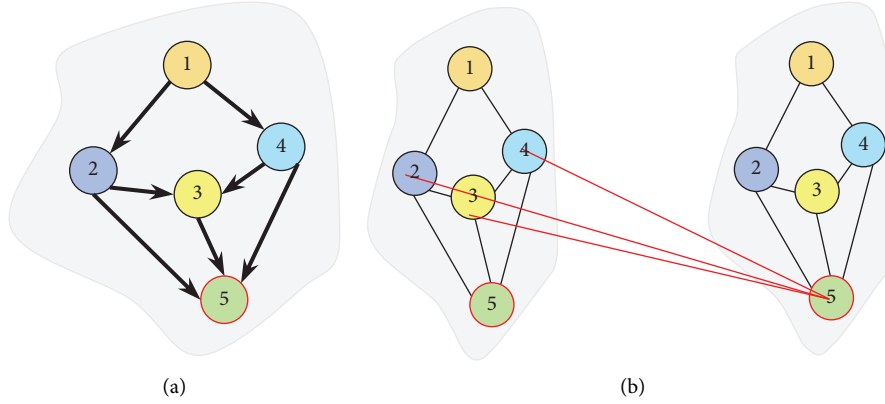


FIGURE 3: Spatial feature modeling.

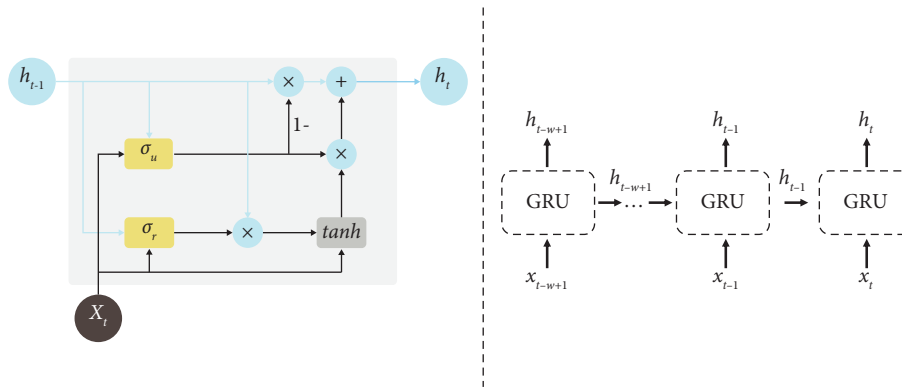


FIGURE 4: Temporal feature modeling.

gates in GRU,  $\mathbf{g}$  controls the degree to which spatial and temporal features contribute to the final spatio-temporal features.

$$\begin{aligned} \mathbf{g} &= \sigma(\mathbf{W}_s \mathbf{H}_s + \mathbf{W}_t \mathbf{H}_t + \mathbf{b}_g), \\ \mathbf{H} &= \mathbf{g} \odot \mathbf{H}_s + (1 - \mathbf{g}) \odot \mathbf{H}_t, \end{aligned} \quad (8)$$

where  $\mathbf{H}_s$  represents the extracted spatial features. Suppose the layer of GCN is  $L$ , then  $H_s$  equals to the output feature matrix  $\mathbf{H}_s^{(L)}$  in  $L$ -th layer.  $\mathbf{H}_t$  represents the extracted temporal features. We stack all units' output hidden state to get the final temporal features.  $\mathbf{W}_s$ ,  $\mathbf{W}_t$ , and  $\mathbf{b}_g$  are learnable parameters.

Let  $\mathbf{Y}_t$  and  $\hat{\mathbf{Y}}_t$  denote the real traffic state and the predicted traffic state at time  $t$ , respectively. In the training process, the target is to minimize the error between  $\mathbf{Y}_t$  and  $\hat{\mathbf{Y}}_t$ , and the loss function is defined as

$$loss = \|\hat{\mathbf{Y}}_t - \mathbf{Y}_t\| + \lambda \|\mathbf{W}\|, \quad (9)$$

where the second term is an L2-regularization term that helps avoid the overfitting problem,  $\lambda$  is the regularization parameter, and  $\mathbf{W}$  represents the weighted parameters.

In summary, we propose a lane-level traffic prediction model. On the one hand, our model uses GCN to model the topological structure of the lane network to extract spatial features. On the other hand, we mine traffic information's

dynamic changes to get the spatial features with GRU model. Finally, we fuse these features to make multi-step traffic predictions.

## 5. Experiment

In this section, we conduct extensive experiment on a real-world dataset to evaluate the proposed method and several baselines answering the following questions:

- (i) RQ1: How does our method perform compared to baselines in lane-level traffic prediction, and do different components of our method show real improvement?
- (ii) RQ2: How do various hyperparameters affect the performance of our method?
- (iii) RQ3: Does our method really work in real situations?

**5.1. Dataset.** We perform experiments on a real-world dataset. The dataset is collected from Sutai Expressway captured by the remote microwaves traffic sensors in Zhejiang, China. Affected by the distance between sensors, we selected 30 sensors' traffic data for experiment. The frequency of traffic information collection for each sensors is once every 5 minutes. The collected data mainly includes

lane-level traffic speed and traffic flow. All traffic information is the average value that passes through the sensor over a sampling time interval. Since some sensors have serious data missing in some lanes, the number of valid lanes in 30 sensors is 53. The record number for each lane is 16,032 and the entire dataset includes 849,696 records in total. To make the prediction result more reliable, we filled in the missing data with the data from the previous moment.

**5.2. Baselines.** To validate the effectiveness of our model for lane-level traffic prediction, we compare the proposed model with several commonly used baselines:

- (1) MLP, which contains three fully connected layers. We concatenate all lanes' traffic information and flatten it, then we input MLP with the flattened information. With the power of nonlinear modeling, MLP can model the spatio-temporal features to some extent.
- (2) LSTM [29], which is a variant of RNN. It achieves efficient modeling of sequence data with gating mechanism. It can only capture temporal features for traffic prediction.
- (3) GRU [30], which is also a variant of RNN and is good at modeling sequence data as well. Compared to LSTM, it has simpler structure. See section for more details. It can only capture temporal features for traffic prediction.
- (4) GCN [38], which can aggregate neighbor information via convolution operations on graph. See Section for more details. It can only capture the spatial features for traffic prediction.
- (5) T-GCN [10], which captures spatial features with GCN and captures temporal features with GRU. Compared to our method, T-GCN constructs a static adjacent matrix and inputs the learned spatial features for spatio-temporal features modeling. It can model the spatio-temporal features for traffic prediction.

**5.3. Implementation and Metrics.** We implement our method with Pytorch and optimize it using the Adam optimizer. The history sequence length for prediction is 12 which means we use the last 60 minutes' traffic data for traffic prediction. The batch size is set to 32. The GCN layer is set to 1. We perform searches for learning rate in  $[1e-2, 1e-3, 1e-4, 1e-5]$ , and choose the learning to be  $1e-4$  finally. In the experiments, the data in all lanes are divided into two parts, the first 80% of the data is used for training and the rest 20% for testing.

To compare the performance of our method and other baselines, we evaluated the prediction results with three widely used metrics.

- (i) Mean absolute error (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i|, \quad (10)$$

- (ii) Root mean squared error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2}, \quad (11)$$

- (iii) Explained variance score (VAR):

$$\text{VAR} = 1 - \frac{\text{Var}\{\mathbf{y} - \hat{\mathbf{y}}\}}{\text{Var}(\mathbf{y})}, \quad (12)$$

where  $\hat{y}_i$  is the predicted value and  $y_i$  is the real value. Var means to calculate the variance. Specifically, for MAE and RMSE, the smaller the value, the better the performance, since they measure the error between the predicted value and the real value directly. VAR measures the quality of the prediction result with calculating the correlation coefficient. The closer the value of VAR is to 1, the better.

**5.4. RQ1: Experiment Results.** Table 2 shows the performance of different models on the prediction of traffic flow of 5 minutes, 10 minutes and 15 minutes using the last 60 minutes traffic flow information, where "ours w/o gate" means that spatial and temporal features are directly added in the feature fusion step without using the gating mechanism in our model. From the results, we have the following observations.

**5.4.1. Observations on Our Method.** Our model consistently works the best compared to all the baselines, which illustrates the superiority of our method for lane-level traffic prediction. (1) Our model performs better than LSTM and GRU. The main reason is that LSTM and GRU can only model the temporal features while they are incapable of capturing spatial features. The results validate the importance of spatial feature modeling in traffic prediction tasks. (2) Our model obtains better results than GCN. Similarly to the reason that our model is better than GRU and GCN, GCN models the spatial features only while it ignores the fact that the traffic data have a significant sequence structure. This means that GCN lacks the ability to model temporal features. The results verify the importance of temporal feature modeling. (3) Our model outperforms T-GCN. One possible reason is that T-GCN is designed for road-level traffic prediction. When faced with complex application scenarios such as lane-level traffic prediction, it is difficult for T-GCN to adopt to such changes, which also verifies the rationality of our model's change for lane-level traffic prediction. Besides, the dataset only contains 53 lanes' traffic information, which means the constructed graph only has 53 nodes. T-GCN performs poorly on such a graph while our model performs well, which further verifies the adaptability of our model to special scenarios. (4) Our model also outperforms MLP. The major reason is that though MLP can model spatio-temporal features to some extent, while it mines the spatio-temporal features roughly and our model can capture rich and effective spatio-temporal features for

TABLE 2: The performance of our model and other baselines on traffic flow prediction. Bold scores are the best and underlined scores are the second best.

T	5 min			10 min			15 min		
Model	MAE	RMSE	VAR	MAE	RMSE	VAR	MAE	RMSE	VAR
MLP	3.4547	4.3616	0.5305	3.4033	4.4673	0.4902	3.7487	4.7843	0.4248
LSTM	3.2121	4.1408	0.5687	3.3041	4.2225	0.5526	3.3171	4.2376	0.5494
GRU	3.1250	4.1229	0.5659	3.3186	4.2368	0.5527	3.3297	4.2511	0.5479
GCN	4.8086	5.9925	0.0836	4.8028	5.9863	0.0855	4.8052	5.9799	0.0876
T-GCN	4.4467	5.4664	0.2541	4.5756	5.6031	0.2186	4.5271	5.6392	0.1879
Ours w/o gate	2.8311	3.7730	0.6380	2.9249	3.8593	0.6221	3.0161	3.9491	0.6049
Ours	<b>2.8274</b>	<b>3.7697</b>	<b>0.6385</b>	<b>2.9168</b>	<b>3.8522</b>	<b>0.6234</b>	<b>2.9951</b>	<b>3.9268</b>	<b>0.6093</b>

traffic prediction. (5) Our method without the gating mechanism performs worse than the full method. The results validate the effectiveness of using the gating mechanism for feature fusion. However, we find that the gating mechanism brings about a slight improvement. One possible reason is that due to the special characteristic of the dataset, spatial features are the main component in the final spatio-temporal features. Different feature fusion methods have little effect on the composition of spatio-temporal features. (6) Our model consistently achieves the best performance when the prediction horizon ranges from 5 minutes to 10 15 minutes. The results indicate that robustness of our method. In addition to the short-term lane-level traffic prediction, it may also be used for long-term lane-level traffic prediction tasks.

5.4.2. *Other Observations.* (1) The performance of LSTM and GRU on all prediction horizons has little difference, although GRU has a simpler structure than LSTM. With limited resources, it is more reasonable to choose GRU instead of LSTM to capture spatial features. (2) GRU obtains better results than GCN. The main reason is that, limited by the number of lanes in the dataset, the ability of GCN to mine spatial features is restricted, and the spatial features captured by GCN are roughly. Also, Traffic data is presented in the form of a sequence, which is naturally more suitable for temporal feature mining. (3) T-GCN outperforms GCN. The reason is that the T-GCN can capture not only the spatial features but also the temporal features, while GCN can only capture the spatial features. The results further verify that both the spatial features and temporal features play an important role in traffic prediction tasks. (4) GRU performs better than T-GCN, though it mines temporal only. One possible reason is that the mechanism for T-GCN to obtain spatio-temporal features is not perfect. When T-GCN is used for lane-level traffic prediction, the extracted spatial features are coarse and inhibit the ability of T-GCN instead.

5.5. *RQ2: Parameter Sensitivity Analysis.* In this part, we conduct sensitivity analysis of two important hyperparameters in our model. The two hyperparameters are the input sequence length  $p$  and the parameter  $\alpha$  that controls the degree to which the dynamic correction matrix  $A_c$  contributes to the data-driven adjacent matrix  $\hat{A}$ .

5.5.1. *Sensitivity of Input Sequence Length  $p$ .* As the input sequence length  $p$  depicts the amount of history traffic information used in traffic prediction, which will be constructed as the initial input features of the model. It is necessary to control the amount of history traffic information for traffic prediction tasks. Specifically, we run our model with the input sequence length  $p$  from 12 to 72. The results of the experiment are shown in Figure 5. The performance of our method decreases when the input sequence length  $p$  increases from 12 to 36, then it increases conversely when  $p$  keeps growing. Overall, the performance remains relatively stable. From the results we can find that further considering longer sequences as input does not bring additional benefit, even a drop has occurred. There exist two possible reasons. (1) Longer sequences may introduce more noise and will limit the predictive power of the model. (2) What we mainly do is short-term traffic prediction and short-term traffic state fluctuates more than long-term traffic state. Shorter input sequences may better help to extract dynamic changes in the data. Such results illustrate that the input sequence length for traffic prediction must be chosen properly. Therefore, the input sequence is supposed to contain sufficient traffic information when the input sequence length is set to 12.

5.5.2. *Sensitivity of  $\alpha$ .* The data-driven adjacent matrix  $\hat{A}$  is composed of distance-based adjacent matrix  $A_d$  and dynamic correlation matrix  $A_c$ . We use parameter  $\alpha$  to control how much  $A_c$  contributes to  $\hat{A}$ . It is also necessary to find out the relationship between the performance of our method and the parameter  $\alpha$ . The input sequence length is set to 12. As shown in Figure 6, our method performs best when  $\alpha = 0.1$ . As  $\alpha$  increases from 0 to 0.1, the performance improves. The main reason is that when  $\alpha$  is too small,  $A_c$  has little effect on  $\hat{A}$ . With increasing  $\alpha$  the influence of  $A_c$  is gradually released to compensate for the static defect caused by the distance-based adjacent matrix, and the performance of the model becomes better. The results verify the necessity and validity of modeling a data-driven adjacent matrix. When  $\alpha$  is larger than 0.1, the performance starts to drop. The possible reason is that larger  $\alpha$  makes the model focus more on the dynamic part of the adjacent matrix and starts to undermine the fundamental role of the static distance-based adjacent matrix. Finally, we set  $\alpha$  to be 0.1.



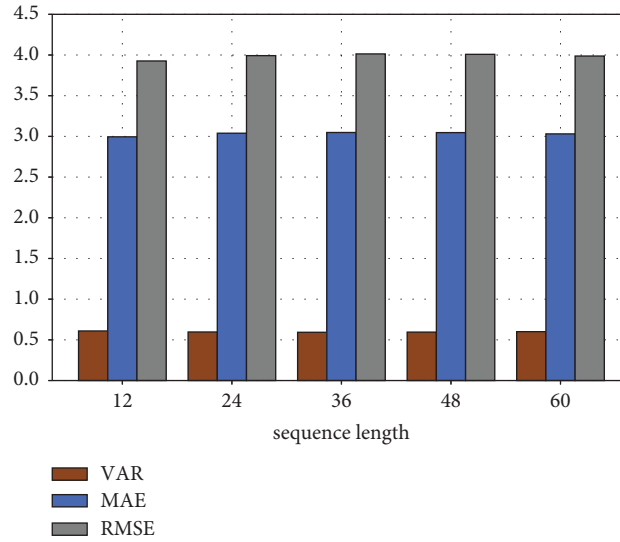


FIGURE 5: The performance of our model as the input sequence length changes.

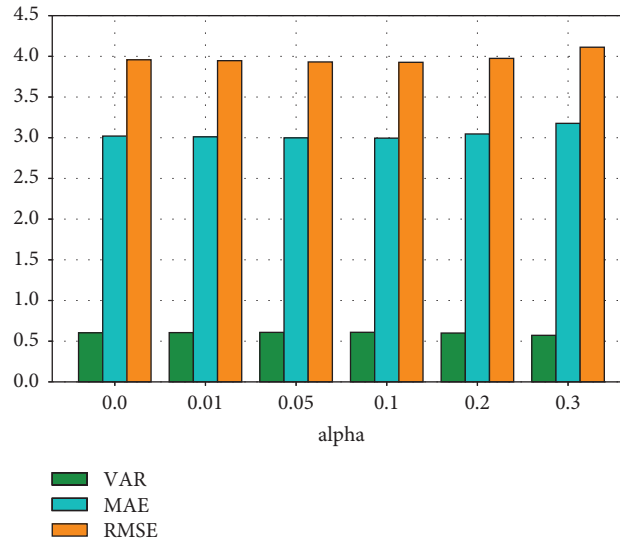


FIGURE 6: The performance of our model as hyperparameter  $\alpha$  changes.

5.6. RQ3: *Model Interpretation*. In this part, we aim to answer RQ3 by visualizing the adjacent matrix and model prediction results.

5.6.1. *Visualization of Adjacent Matrix*. To figure out how the data-driven adjacent matrix affects the prediction results of the model, we visualize the distance-based adjacent matrix  $A_d$  and lane correlation matrix  $A_c$  in the morning, afternoon, and evening of a random day, respectively. As shown in Figure 7, the distance-based adjacent matrix and the correlation matrix show different relationship patterns between lanes in different situations. More specifically,  $A_d$  is static and  $A_c$  presents different data distributions at different times. It can be seen from the answer of RQ3 that the construction of the dynamic correlation matrix plays

a positive role in the prediction of the model. The reason is that the traffic information at different times changes dynamically, and it is difficult for the static distance-based adjacent matrix to respond to such dynamic changes in a timely and effective manner. With the help of the dynamic correlation matrix, the data-driven adjacent matrix has acquired the ability to model dynamically changing traffic information. Further, the performance of the model can be improved.

5.6.2. *Visualization of Prediction Result*. To better understand our proposed method, we randomly selected 4 lanes in different sections of the road and visualize the ground truth of the test set and the prediction result. As shown in Figure 8, the results show the following:

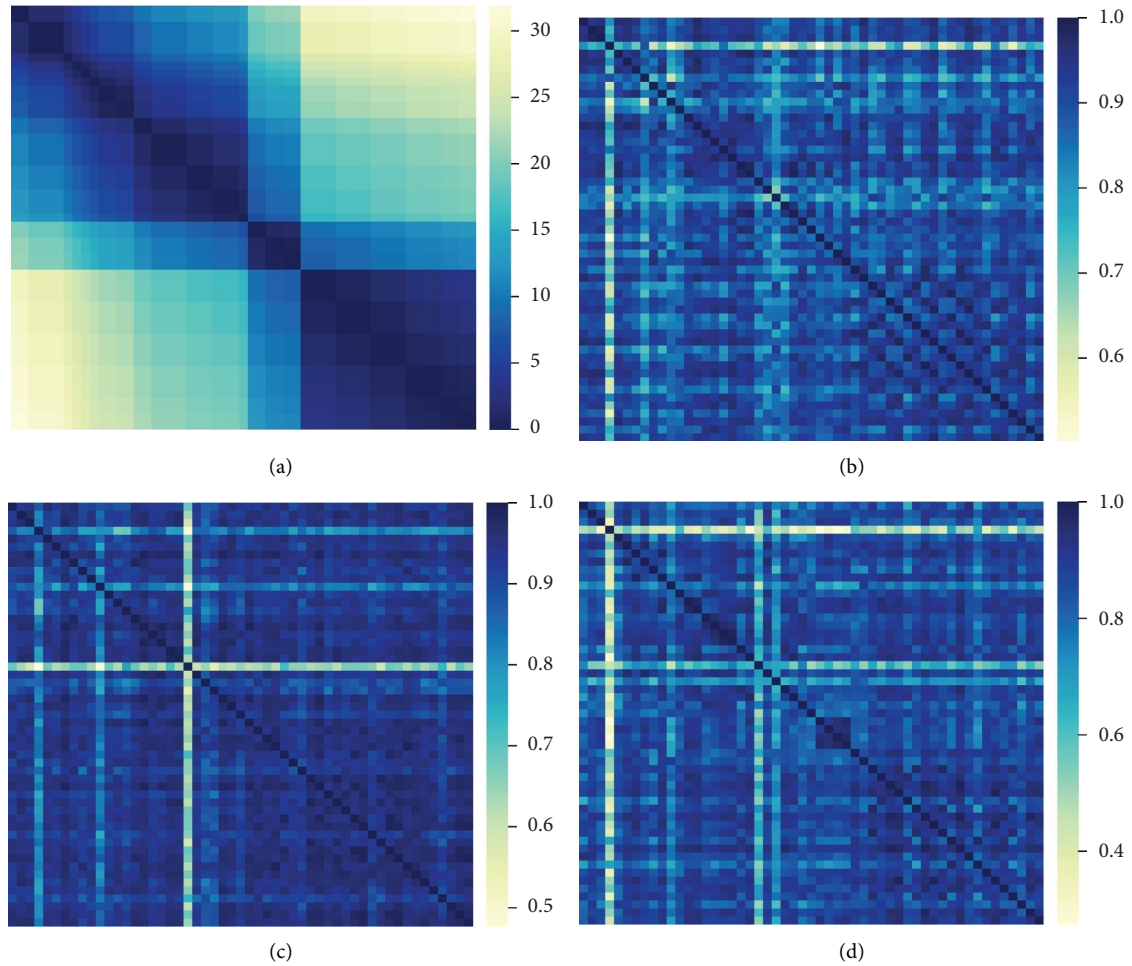


FIGURE 7: Visualization of distance-based adjacent matrix  $A_d$  and dynamic correlation matrix  $A_c$ . (a) Visualizes the distance-based adjacent matrix. (b) Depicts the correlation matrix of traffic flow collected from 8 am to 9 am. (c) The correlation matrix of traffic flow collected from 6 pm to 7 pm. (d) The correlation matrix of traffic flow collected from 10 pm to 11 pm. (a–c) represent the similarity relationship of traffic flow in different lanes in the morning, afternoon, and evening of a random day respectively.

- (1) Similar patterns exist between the ground truths of different lanes. The geographic distance between some lanes may be too large, in which case the distance-based adjacent matrix ignores the similar pattern. As a result, the lanes that are too far apart cannot cooperate with each other to improve the final prediction quality. Our proposed data-driven adjacent matrix effectively alleviates this problem on the basis of distance-based adjacent matrix.
- (2) There is a certain error between the prediction results of the model and the ground truth. The reasons are threefold: (a) GCN defines a smooth filter and models the spatial features with the filter on the spatial domain. This process of aggregating neighbor information results in smooth predictions. (b) Though the construction of data-driven adjacent matrix can help to improve model's performance. However, the

use of the correlation matrix enables GCN to aggregate more information from more neighbors, which further leads to a smoother prediction result of the model. (c) When the ground truth is small, the same gap can cause a larger relative error.

- (3) Our model can capture the traffic trend at the lane level. This property can help formulate effective and detailed traffic control strategies in real time and realize scientific traffic management planning.

Besides, Figure 9 displays the ground truth and prediction results for a randomly chosen weekend and a randomly chosen weekday. As depicted in Figure 9, our model can capture trends of traffic flow throughout the day. Whether it is weekdays or weekends, peak or low-peak times, traffic flow trends are well modeled, allowing for our model to realize stable and reliable prediction.

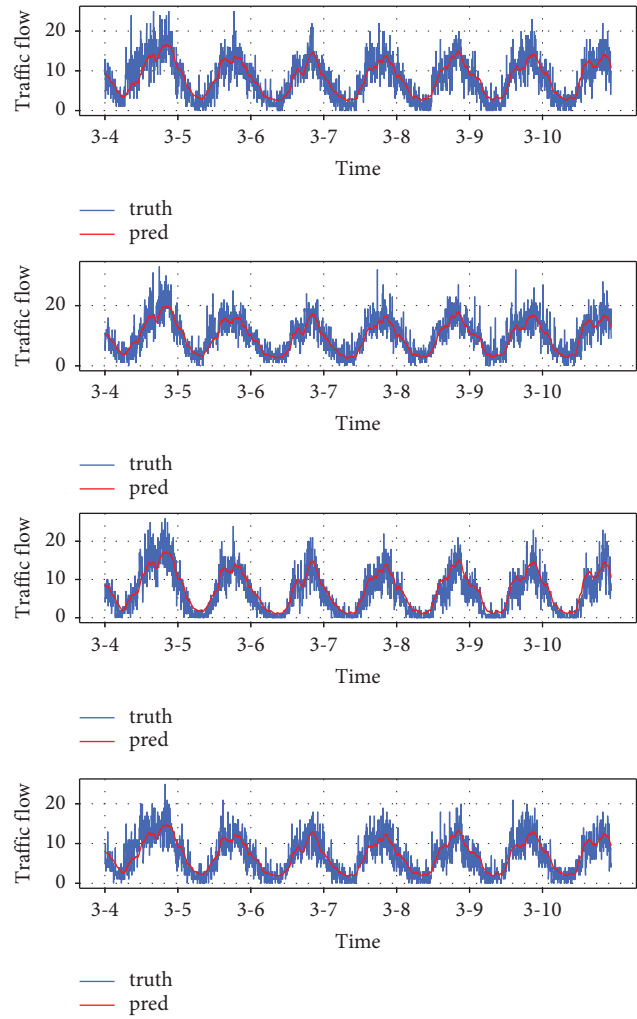


FIGURE 8: Visualization of ground truth and prediction results of 4 randomly selected lanes on the test set. The blue line represents the ground truth and the red line represents the prediction results. The horizontal axis represents the traffic flow, and the vertical axis represents the time the traffic flow recorded or predicted.

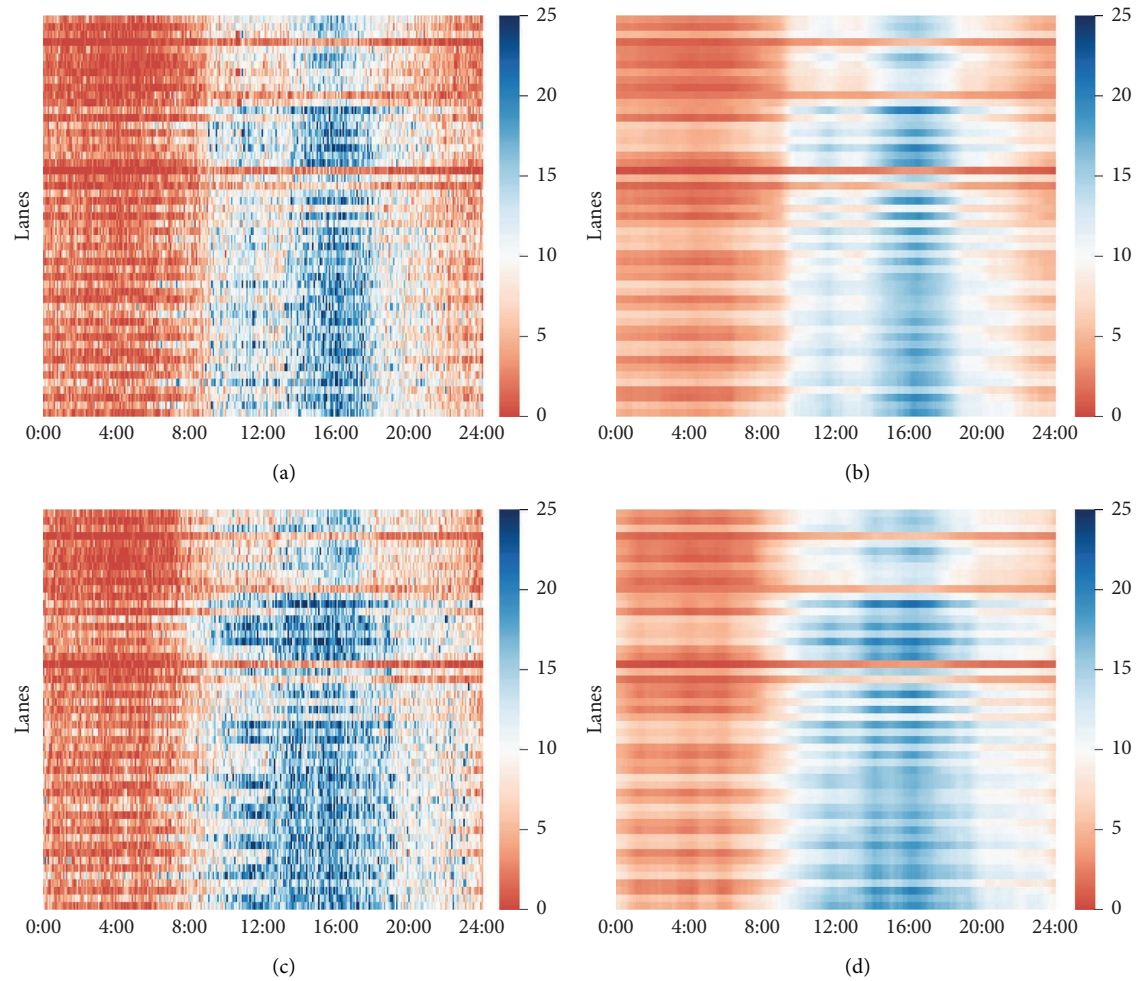


FIGURE 9: Visualization of ground truth and prediction results of all lanes on the test set. (a) Ground truth on a weekend. (b) Prediction results on the same weekend as (a). (c) Ground truth on a weekday. (d) Prediction results on the same weekday as (c).

## 6. Conclusion

In this paper, we propose a lane-level traffic prediction model for lane-level traffic prediction tasks. Specifically, we capture spatial features with GCN. To adapt to the lane-level traffic prediction, we treat different lanes of the same road segment as different nodes on the graph. Furthermore, considering that most existing graph-based methods build static adjacent matrix, we construct a data-driven adjacent matrix, which consists of a static distance-based adjacent matrix and a dynamic lane correlation matrix. Additionally, we utilize GRU to capture temporal features. Then we adaptively fuse spatial features and temporal features through the gating mechanism to obtain spatio-temporal features for multi-step lane-level traffic prediction. Experiment on a real-world dataset verified the effectiveness of our model for lane-level traffic prediction.

In the future, we plan to consider more auxiliary features, such as utilizing speed information in traffic flow prediction. The current model can also be improved by considering more detailed and realistic lane relationships for

more than two lanes. Besides, we consider to combine with some advanced techniques such as attention network to achieve more accurate and reliable lane-level traffic prediction.

## Data Availability

The data used during the study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China under Grant no. 62202131, the Natural Science Foundation of Zhejiang Province under no. LQ20F020015, and the Key Science and Technology Project of Zhejiang Province under no. 2020C01165.

## References

- [1] C. Chen, Q. Liu, X. Wang, C. Liao, and D. Zhang, "Semi-traj2graph identifying fine-grained driving style with gps trajectory data via multi-task learning," *IEEE Transactions on Big Data*, vol. 8, no. 6, pp. 1550–1565, 2021.
- [2] C. Chen, L. Li, and M. Li, "A generic framework for bi-criteria optimum path-finding based on deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1–13, 2022.
- [3] L. Jiang, C.-X. Chen, and C. Chen, "L2mm: learning to map matching with deep models for low-quality gps trajectory data," *ACM Transactions on Knowledge Discovery from Data*, vol. 17, no. 3, pp. 1–25, 2023.
- [4] J. Zhao, C. Chen, C. Liao et al., "2f-tp: learning flexible spatiotemporal dependency for flexible traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 1–13, pp. 1–13, 2022.
- [5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 1–9, 2014.
- [6] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, no. 3–19, pp. 3–19, 2014.
- [7] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.
- [8] S. Cheng, F. Lu, P. Peng, and S. Wu, "Short-term traffic forecasting: an adaptive st-knn model that considers spatial heterogeneity," *Computers, Environment and Urban Systems*, vol. 71, pp. 186–198, 2018.
- [9] M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong, and L. D. Han, "Online-svr for short-term traffic flow prediction under typical and atypical traffic conditions," *Expert Systems with Applications*, vol. 36, no. 3, pp. 6164–6173, 2009.
- [10] L. Zhao, Y. Song, C. Zhang et al., "T-GCN: a temporal graph convolutional network for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [11] K. Lee and W. Rhee, "Ddp-gcn: multi-graph convolutional network for spatiotemporal traffic forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 134, Article ID 103466, 2022.
- [12] Y. Ge, J. F. Zhai, and P. C. Su, "Traffic flow prediction based on multi-spatiotemporal attention gated graph convolution network," *Journal of Advanced Transportation*, vol. 2022, Article ID 2723101, 9 pages, 2022.
- [13] D. Tian, G. Wu, P. Hao, K. Boriboonsomsin, and M. J. Barth, "Connected vehicle-based lane selection assistance application," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2630–2643, 2019.
- [14] Z. Yao, R. Hu, Y. Wang, Y. Jiang, B. Ran, and Y. Chen, "Stability analysis and the fundamental diagram for mixed connected automated and human-driven vehicles," *Physica A: Statistical Mechanics and its Applications*, vol. 533, Article ID 121931, 2019.
- [15] C. F. Daganzo, "A behavioral theory of multi-lane traffic flow. part i: long homogeneous freeway sections," *Transportation Research Part B: Methodological*, vol. 36, no. 2, pp. 131–158, 2002.
- [16] C. F. Daganzo, "A behavioral theory of multi-lane traffic flow. part ii: merges and the onset of congestion," *Transportation Research Part B: Methodological*, vol. 36, no. 2, pp. 159–169, 2002.
- [17] P. G. Michalopoulos, D. E. Beskos, and Y. Yamauchi, "Multilane traffic flow dynamics: some macroscopic considerations," *Transportation Research Part B: Methodological*, vol. 18, no. 4-5, pp. 377–395, 1984.
- [18] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model," in *Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV)*, pp. 607–612, IEEE, Baden-Baden, Germany, July 2011.
- [19] B. M. Williams, "Multivariate vehicular traffic flow prediction: evaluation of arimax modeling," *Transportation Research Record*, vol. 1776, no. 1, pp. 194–200, 2001.
- [20] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, "Short-term prediction of lane-level traffic speeds: a fusion deep learning model," *Transportation Research Part C: Emerging Technologies*, vol. 106, no. 1–16, pp. 1–16, 2019.
- [21] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.
- [22] K. Y. Chan, T. S. Dillon, J. Singh, and E. Chang, "Neural-network-based models for short-term traffic flow forecasting using a hybrid exponential smoothing and levenberg-marquardt algorithm," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 2, pp. 644–654, 2012.
- [23] C. P. I. J. Van Hinsbergen, T. Schreiter, F. S. Zuurbier, J. W. C. Van Lint, and H. J. van Zuylen, "Localized extended kalman filter for scalable real-time traffic state estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 1, pp. 385–394, 2012.
- [24] M. G. Karlaftis and E. Vlahogianni, "Statistical methods versus neural networks in transportation research: differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.
- [25] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.
- [26] W. Lu, Y. Rui, Z. Yi, B. Ran, and Y. Gu, "A hybrid model for lane-level traffic flow forecasting based on complete ensemble empirical mode decomposition and extreme gradient boosting," *IEEE Access*, vol. 8, pp. 42042–42054, 2020.
- [27] I. Lana, J. L. Lobo, E. Capecci, J. Del Ser, and N. Kasabov, "Adaptive long-term traffic state estimation with evolving spiking neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 126–144, 2019.
- [28] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowledge-Based Systems*, vol. 172, no. 1–14, pp. 1–14, 2019.
- [29] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [30] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, Doha, Qatar, October 2014.
- [31] J. Yang, J. Li, L. Wei, L. Gao, and F. Mao, "St-agrnn: a spatio-temporal attention-gated recurrent neural network for traffic state forecasting," *Journal of Advanced Transportation*, vol. 2022, Article ID 2806183, 17 pages, 2022.

- [32] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.
- [33] Y. Jia, J. Wu, and M. Xu, "Traffic flow prediction with rainfall impact using a deep learning method," *Journal of Advanced Transportation*, vol. 2017, Article ID 6575947, 10 pages, 2017.
- [34] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: a path based deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 372–385, 2019.
- [35] S. Deng, S. Jia, and J. Chen, "Exploring spatial-temporal relations via deep convolutional neural networks for traffic flow prediction with incomplete data," *Applied Soft Computing*, vol. 78, pp. 712–721, 2019.
- [36] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact," *Transportation Research Record*, vol. 2674, no. 4, pp. 459–470, 2020.
- [37] Q. Liu, B. Wang, and Y. Zhu, "Short-term traffic speed forecasting based on attention convolutional neural network for arterials," *Computer-Aided Civil and Infrastructure Engineering*, vol. 33, no. 11, pp. 999–1016, 2018.
- [38] N. Thomas and K. M. Welling, "Semi-supervised Classification with Graph Convolutional Networks," in *Proceedings of the 5th International Conference on Learning Representations*, Vienna, Austria, January 2017.
- [39] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: data-driven traffic forecasting," in *Proceedings of the 6th International Conference on Learning Representations*, California, CA, USA, July 2018.
- [40] B. Yu, H. Yin, and 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*, pp. 3634–3640, Stockholm, Sweden, July 2018.
- [41] K. Guo, Y. Hu, Y. Sun, S. Qian, J. Gao, and B. Yin, "Hierarchical graph convolution network for traffic forecasting," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 1, pp. 151–159, 2021.
- [42] K. Zhu, S. Zhang, J. Li, D. Zhou, H. Dai, and Z. Hu, "Spatiotemporal multi-graph convolutional networks with synthetic data for traffic volume forecasting," *Expert Systems with Applications*, vol. 187, Article ID 115992, 2022.
- [43] K. Guo, Y. Hu, Z. Qian, Y. Sun, J. Gao, and B. Yin, "Dynamic graph convolution network for traffic forecasting based on latent network of laplace matrix estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1009–1018, 2022.
- [44] Y. Cao, D. Liu, Q. Yin, F. Xue, and H. Tang, "Msasgcn: multi-head self-attention spatiotemporal graph convolutional network for traffic flow forecasting," *Journal of Advanced Transportation*, vol. 2022, Article ID 2811961, 15 pages, 2022.
- [45] Y. Ma, Z. Zhang, and A. Ihler, "Multi-lane short-term traffic forecasting with convolutional lstm network," *IEEE Access*, vol. 8, pp. 34629–34643, 2020.
- [46] W. Lu, Y. Rui, and B. Ran, "Lane-level traffic speed forecasting: a novel mixed deep learning model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 4, pp. 3601–3612, 2022.
- [47] L. Wang, G. Shen, K. Yu, Z. Ji, and X. Kong, "Lane-level traffic flow prediction based on dynamic graph generation," in *Proceedings of the 2021 IEEE 23rd Int Conf on High Performance Computing & Communications; 7th Int Conf on Data Science & Systems; 19th Int Conf on Smart City; 7th Int Conf on Dependability in Sensor, Cloud & Big Data Systems & Application (HPCC/DSS/SmartCity/DependSys)*, pp. 1802–1809, IEEE, Hainan, China, December 2021.
- [48] Z. Li, G. Xiong, Y. Tian et al., "A multi-stream feature fusion approach for traffic prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 2, pp. 1456–1466, 2022.
- [49] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and 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*, vol. 30, no. 3, pp. 83–98, 2013.
- [50] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: a survey," *Expert Systems with Applications*, vol. 207, Article ID 117921, 2022.