IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

IEEE VEHICULAR TECHNOLOGY SOCIETY SECTION

Received 24 June 2023, accepted 11 August 2023, date of publication 24 August 2023, date of current version 6 September 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3308238*

RESEARCH ARTICLE

# A Graph Deep Learning-Based Fast Traffic Flow Prediction Method in Urban Road Networks

**DONGFANG YANG AND LIPING LV**
School of Information Engineering, Zhengzhou Shengda University, Zhengzhou 451191, China
Corresponding author: Dongfang Yang (100603@shengda.edu.cn)

This work was supported by the Key Projects of Science and Technology of Henan Province under Grant 222102210209.

**ABSTRACT** In modern smart cities, road networks are becoming more and more complicated, resulting in more complex format of graphs. This brings many challenges to the forecasting of traffic flow in road graphs. Most of traditional traffic flow forecasting methods ignored many implicit relationships inside road graphs. And this cannot be well suitable for modern road networks in smart cities. Besides, the operation of smart cities is accompanied with real-time big data stream. The running efficiency of forecasting methods is another important concern. To handle this issue, this paper proposes a graph deep learning-based fast traffic flow forecasting method in urban road networks. Firstly, the theory about graph convolution operations is deduced and can be used as the basis of a graph convolution network (GCN). Then, the whole road network is viewed as a complex road graph, and the GCN is introduced to establish a novel forecasting method for graph-level traffic flow. With roads being regarded as nodes and their relations being regarded as edges, graph-level forecasting can be realized with the use of the proposed method. Experiments are carried out on a standard real dataset to evaluate the proposal. The experimental results show a proper performance of the proposal.

**INDEX TERMS** Graph deep learning, road traffic prediction, fast prediction, traffic flow, road networks.

## I. INTRODUCTION

In contemporary society, the continuous progress of industry has brought huge resources and wealth to the society. With the gradual popularization of cars in general families, urban roads all over the world are facing huge traffic flow pressure [1], [2]. Urban road networks are the lifeblood of urban economic development, and their carrying capacity determines the efficiency of economic development [3], [4]. Many countries have also taken some policy interventions to cope with the increasing traffic management pressure, such as traffic limitation and green travel promotion [5]. However, this still cannot well solve the problem, because the base amount of car ownership is always very large [6], [7]. In recent years, various countries have begun to explore the use of technical means to build some intelligent traffic management systems [8], [9]. Among them, the ability to accurately predict the traffic flow values in the future is an important prerequisite for building an intelligent traffic management system [10].

The associate editor coordinating the review of this manuscript and approving it for publication was Hassan Omar.

As is stated by related works such as [11] and [12], traffic flow forecasting is of great significance to construction of intelligent transportation systems. Results delivered by such studies can be used in numerous detailed follow-up analyses, but also to make planning decisions and to design transport infrastructure as well as traffic control systems [13]. Besides, for the area of road maintenance engineering, various traffic flow values in different roads directly influence the usage loss status of roads. If the road management departments can predict the traffic flow of the roads in advance, it will be beneficial to their future affair planning. To this end, the traffic flow forecasting is a significant work in many areas, and this study concentrates on the smart methods that can be utilized to predict future traffic flow values [14].

In the past decade, traffic flow prediction has become a common concern in academic circles [15], [16], [17]. Machine learning has become an important means to solve the traffic flow prediction problem [18], [19], [20]. Through establishing statistical models to find the potential laws in historical data, the prediction results of future traffic flow can be calculated [21], [22], [23], [24], [25], [26], [27], [28], [29].

Although machine learning has made great progress in traffic flow prediction technology in recent years, there are still some limitations that have not been solved [30], [31], [32]. Because the existing machine learning prediction models focus on the pattern rules at the data level, thus ignoring the internal structural characteristics of the road network. In fact, as is shown in Fig. 1, urban road network can be viewed as a kind of complex network structure, which contains different nodes and complex relationships. The structural characteristics of road networks have a great impact on the formation of traffic flow. Ignorance of this point will make it difficult for the prediction model to have good generalization performance.

In order to deal with this problem, this paper uses the graph deep learning method to represent the road networks. Graph deep learning is the fusion of graph theory and deep learning. Because graph is an important data representation format, deep learning can be applied to the data with graph structure to obtain more in-depth feature representation of objects. In the road network scenes, the method of graph deep learning can be used to capture the structural characteristics of the road network, so as to establish a powerful prediction model. Therefore, this paper proposes a graph deep learning-based traffic flow forecasting model for urban road networks (named as DTFUN for short). Specifically, each road is regarded as a node, and the association between roads is regarded as the edge between nodes, so as to construct a graph network with graph structure. Then, the graph convolution network is used to model the structural characteristics of the above graph network, so as to obtain the forecasted results for the subsequent traffic flow values. The main contributions of this paper can be summarized in three aspects:

- This paper declares that traffic flow forecasting needs to employ structural characteristics in road networks.
- The graph deep learning is employed to construct a strong forecasting model for traffic flow forecasting.
- The experiments are carried out on a real dataset, so as to evaluate practivity of the proposal.

The rest of this paper is divided into several sections. In Section II, the problem scenario is stated and some related works are surveyed. In Section III, the main technical methodology is described. In Section IV, experiments are conducted and the obtained results are analyzed, so that performance of the proposal can be evaluated. In Section V, this paper is summarized and concluded.

## II. PRELIMINARIES

### A. RELATED WORK

The task of traffic flow prediction has always been an important issue in traffic management and planning, and significant achievements have been made in the past few decades. In recent years, with the development and application of deep learning methods, research on traffic flow prediction based on deep learning methods has gradually become a hot topic. Previous traffic flow prediction methods mostly processed traffic flow data based on specific scenarios, ignoring the spatial correlation of road networks. At present, existing methods mainly focus on the complex spatial structure of traffic road networks and the temporal dependence of flow data, and expand research on the basis of spatiotemporal characteristics.

Zhao et al. [33] proposed the temporary graph convolutional network (T-GCN) model, which explores the performance of traffic prediction from both temporal and spatial dimensions. T-GCN uses GCN to learn the Space complexity between traffic roads, and GRU model to learn the time series dependency of traffic data.

Wang et al. [34] proposed a spatiotemporal graph neural network for dynamic prediction of traffic flow. The spatial layer of this model is used to extract spatial relationships between traffic networks, the GRU layer is used to extract local temporal correlations of traffic data, and the Transformer layer is used to directly learn global temporal features in the data sequence.

Wang et al. [35] proposed an ST-GCN method that can predict traffic flow without historical data. On the basis of extracting spatiotemporal features using GCN and GRU models, this method incorporates the Adjacent Similar algorithm to predict traffic flow at intersections without historical data.

Lai et al. [36] introduced the NodeRank algorithm to calculate the importance of road nodes based on extracting the spatiotemporal features of traffic flow prediction tasks.

Qi et al. [37] proposed an asynchronous graph convolutional networks (FedAGCN) based on joint learning, starting from the accuracy and time cost of traffic flow prediction. This method designs a cloud model to aggregate the global parameters of each submodel. The whole learning task is divided into several sub graph Learning space features, and joint learning is used to retain the local correlation of parameter updates.

Zheng et al. [38] proposed an STA-ED framework based on the scenario of predicting the flow of different vehicle models on the traffic network. This method sequentially inputs traffic data into the Spatial Attention Layer, LSTM Encoder, Temporary Attention Layer, LSTM Decoder, and finally obtains traffic prediction values.

Duan et al. [39] focused on the spatial, temporal, and prediction cycle dimensions of traffic flow prediction. By improving the graph attention network, dynamic prediction of traffic flow was achieved.

The typical existing research works are summarized in TABLE 1.

### B. PROBLEM STATEMENT

Main workflow of the proposed DTFUN is shown in Fig. 2. It is assumed that the whole road networks are viewed as a graph-structured network, in which each road is a node and relations among roads are edges. Let $i$ denote index number of roads and range from 1 to $I$, and $t$ denote index number of timestamps and range from 1 to $T$. Thus, $x_i$ denotes the $i$-th road node and $X_i^{(t)}$ denotes traffic flow value of $x_i$ at the

**FIGURE 1.** An illustration for the urban intelligent traffic systems.

**TABLE 1.** Summarization of typical existing research works.

| References | Main highlights |
|---|---|
| Zhao et al. [33] | Exploring the performance of traffic prediction from both temporal and spatial dimensions |
| Wang et al. [34] | Using spatial layer to extract spatial relationships between traffic networks |
| Wang et al. [35] | Incorporating the Adjacent Similar algorithm to predict traffic flow at intersections without historical data |
| Lai et al. [36] | Introducing the NodeRank algorithm to calculate the road importance based on spatiotemporal features |
| Qi et al. [37] | Designing a cloud model to aggregate the global parameters of each submodel |
| Zheng et al. [38] | Sequentially inputting traffic data into several neural computing structures to obtain prediction results |
| Duan et al. [39] | Focusing on the spatial, temporal, and prediction cycle dimensions of traffic flow prediction |

$t$-th timestamp. It is assumed that there are $T$ timestamps in training data. Given all the traffic flow values of the $T$ timestamps, the goal is to train a forecasting model according to the historical data, which can be represented as follows:

$$X_i^{(t)} = f_i\left[x\left(t\right)\right] \tag{1}$$

It represents that the $X_i^{(t)}$ is related to features of $x_i$ at the $t$-th timestamp. When the index number of timestamps $t$ is given, forecasting value of traffic flow at that time can be calculated. From the macroscopic view, this work manages to learn a sequential model that can generate forecasting results for traffic flow values after the $T$-th timestamp. The process can be represented as follows:

$$\underbrace{X_i^{(1)}, \cdots, X_i^{(t)}, \cdots, X_i^{(T)}}_{Training} \Rightarrow \underbrace{X_i^{(T+1)}, X_i^{(T+2)}, \cdots}_{Forecasting} \tag{2}$$

For a road node $x_i$, it has two main levels of features: traffic flow values and adjacency relations. The former represents the inherent traffic flow value of itself, and the latter represents implicit relations between itself and other road nodes. The adjacency can be specifically defined before modeling, and the adjacency relations between $x_i$ and other road nodes are represented using an adjacency matrix. Having integrated the two parts of features together, deep representation for the road node $x_i$ can be obtained. Then, the GCN model can be utilized for modeling and to output forecasting results. The next section is going to present more detailed mathematical description for the GCN modeling process, as well as its optimization and training process.

## III. METHODOLOGY

This section gives mathematical descriptions for GCN model through two subsections. The first subsection gives some basic mathematical preliminaries, and the second subsection gives the main reduction process of the GCN model, as well as its objective function. The variables involved in this section are listed and explained in TABLE 2.

### A. MATHEMATICAL FOUNDATION

The classical Fourier transform in continuous intervals can be defined as the following formula:

$$F\left[x\left(t\right)\right] = \int x\left(t\right) e^{-2\pi j\varpi t} dt \tag{3}$$

where $F\left(\cdot\right)$ denotes the Fourier transform operator, and $j$ denotes the imaginary number and $j^2 = -1$. The Fourier transform manages to map signals in the time domain into the spectral domain, while the inverse Fourier transform manages to map signals in the spectral domain into the time domain.

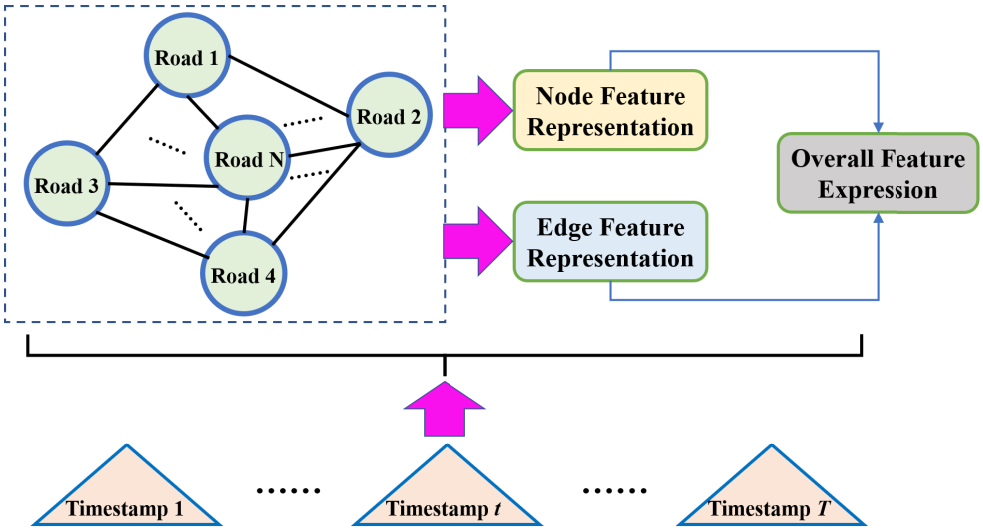



**FIGURE 2.** Major workflow of the proposed DTFUN.

**TABLE 2.** Explanation for variables involved in this paper.

| Variables | Explanation | Notes |
|---|---|---|
| $i$ | Index number of roads | $i = 1, 2, \cdots, I$ |
| $t$ | Index number of timestamps | $t = 1, 2, \cdots, T$ |
| $x_i$ | The $i$th road (also known as the node) | |
| $v_i^{(t)}$ | The traffic flow value of $x_i$ at the $t$th timestamp | |
| $f_i$ | the forecasting model that functions on $x_i$ | |
| $F(\cdot)$ | The Fourier transform function | |
| $F^{(-1)}(\cdot)$ | The reverse Fourier transform function | |
| $x(t)$ | Samples under the $t$th timestamp | |
| $U$ | A matrix used for field transformation | |
| $g$ | The variable used for graph convolution transformation | |
| $\lambda_1, \lambda_2, \cdots, \lambda_n$ | $n$ core parameters of the graph convolution filter | |
| $\beta, \beta_0, \beta_1$ | temporal parameters to be learned | |
| $D$ | degree matrix of a graph network | |
| $A$ | adjacency matrix of a graph network | |
| $E$ | identity matrix | |
| $R_i^{(t)}$ | the predicted traffic flow value of $x_i$ at the $t$th timestamp | |

The inverse Fourier transform can be represented as:

$$x(t) = F^{-1}\{F[x(t)]\} \tag{4}$$

where $F^{-1}(\cdot)$ denotes inverse Fourier transform operator. The inverse Fourier transform in continuous intervals can be represented as the following formula:

$$x(t) = \int F[x(t)] e^{2\pi j \varpi t} dt \tag{5}$$

The Fourier transform has a very important characteristics in terms of convolution. The convolution operations in time domain, the convolution operations will be transformed into multiplication operations when they are mapped into the spectral domain using Fourier transform. The following formula can be deduced:

$$F[x_1(t) * x_2(t)] = F[x_1(t)] \cdot F[x_2(t)] \tag{6}$$

Similarly, an expression about inverse Fourier transform can be also deduced as follows:

$$x_1(t) * x_2(t) = F^{-1}\{F[x_1(t)] \cdot F[x_2(t)]\} \tag{7}$$

It is known that general convolution operations have much computational complexity. The convolution operations can transform signals in time domain into the spectral domain,

so that convolution operations in time domain can be transferred into multiplication operations in spectral domain. After that, calculation results in spectral domain can be transformed into the results in time domain. In other words, the introduction of convolution operations can reduce some computational complexity in convolution operations.

The essence of inverse Fourier transform is to express a function as a linear combination of several orthogonal basis functions. Thus for graph-level signal, the Fourier transform selects eigenvectors of Laplace as the basis function. For a signal in a graph can be denoted as:

$$x = x(\lambda_1) \cdot u_1 + x(\lambda_2) \cdot u_2 + \cdots + x(\lambda_n) \cdot u_n \tag{8}$$

Due to the fact that $U = (u_1, u_2, \cdots, u_n)$ is actually $n$ linear independent vectors in a $n$-dimension space, the Laplace matrix is selected for use. Hence, graph-level Fourier transform and inverse Fourier transform can be represented as the following formulas, respectively:

$$F(x) = U^T x \tag{9}$$

$$x = U \cdot F(x) \tag{10}$$

### B. GRAPH CONVOLUTION NETWORK

The main idea thought of GCN is demonstrated in Fig. 3. Let $x_i$ denote the set of $I$ road nodes, where $i$ ranges from 1 to $I$. In GCN, a basic operation is the graph convolution. The graph convolution between $x_i$ and a graph convolution filter $g$ is represented as:

$$x_i \otimes g = F^{-1}[F(x_i) \odot F(g)] = U\left[U^T x_i \odot U^T g\right] \tag{11}$$

where $\otimes$ denotes the graph convolution operator, and $\odot$ denotes the harmand multiplication. Assuming that $U^T g$ is with the format of a diagonal matrix, the above formula can be rewritten as:

$$x_i \otimes g = UF(g)\, U^T x = U \begin{bmatrix} F(g_1) & & \\ & \ddots & \\ & & F(g_n) \end{bmatrix} U^T x \tag{12}$$

The GCN model can be viewed as a specific example of the above formula.

Expanding the $F(g)$ with use of the first-order Chebyshev polynomial, the $F(g)$ in the above formula can be rewritten as follows:

$$F(g) = \begin{bmatrix} \beta_0 C_0(\lambda_1) + \beta_1 C_1(\lambda_1) & & \\ & \ddots & \\ & & \beta_0 C_0(\lambda_n) + \beta_1 C_1(\lambda_n) \end{bmatrix} \tag{13}$$

where $\lambda_1, \lambda_2, \cdots, \lambda_n$ are core parameters of the graph convolution filter. To further simplify the above formula, the following formula can be deduced:

$$x_i \otimes g = \left[\beta_0 - \beta_1 \left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}}\right)\right] x_i \tag{14}$$

where $\beta_0$ and $\beta_1$ are parameters, $D$ is the degree matrix of the graph, and $A$ is the adjacency matrix of the graph. The parameters in the above formula can be further reduced, so that the above formula can be rewritten as:

$$x_i \otimes g = \beta\left(D^{-\frac{1}{2}} A D^{-\frac{1}{2}} + E\right) x \tag{15}$$

where $\beta$ is the parameter to be learned, and $E$ denotes identity matrix. To ensure the stability of GCN, the above formula can be finally approximated as follows:

$$x_i \otimes g = \beta\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}\right) x \tag{16}$$

where $\tilde{A}$ and $\tilde{D}$ are calculated as follows:

$$\tilde{A} = A + E \tag{17}$$

$$\tilde{D} = \sum \tilde{A} \tag{18}$$

For road node at the $t$-th timestamp, its feature representation can be obtained as follows:

$$R_i^{(t)} = v_i^{(t)} \cdot \beta\left(\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}\right) \tag{19}$$

where $v_i^{(t)}$ denotes traffic flow value of $x_i$ at the $t$-th timestamp. Given all the training data, the learning goal is to search optimal solutions of the following formula:

$$\min_{i,t} \sum_{t=1}^{T} \sum_{i=1}^{I} \left[\left\| R_i^{(t)} - \hat{R}_i^{(t)} \right\| + \alpha \cdot \|\Theta\|\right] \tag{20}$$

where $\hat{R}_i^{(t)}$ denotes real traffic flow value at the $t$-th timestamp, $\alpha$ denotes penalty parameter that needs to be set manually, and $\Theta$ denotes the set of parameters. Then, the adaptive momentum estimation is selected as the optimizer to search solutions. After training, the parameters $\Theta$ can be learned and the forecasting model in Eq. (19) can be formulated.

## IV. EXPERIMENTS AND ASSESSMENT

### A. BASIC SCENARIOS

The dataset is a universal one that is used for evaluation of traffic flow forecasting. It was released by the Caltrans Performance Measurement System, and was thus named as PeMS for short. It has four most typical versions that are named as PeMS03, PeMS04, PeMS07 and PeMS08, respectively. They mainly record traffic flow values in some continuous time intervals. This work uses the version of PeMS03 for evaluation.

For the PeMS03 dataset, its initial data format is with the continuous format. In other words, its data can be visualized as a continuous curve within a continuous interval. The PeMS03 dataset has 358 monitoring nodes, and is involved with the interval from Sep 1, 2018 to Nov 30, 2018. In order to transform the initial continuous values into the discrete forms that can be substituted into models, sampling operations are conducted under some specific frequency. Here,

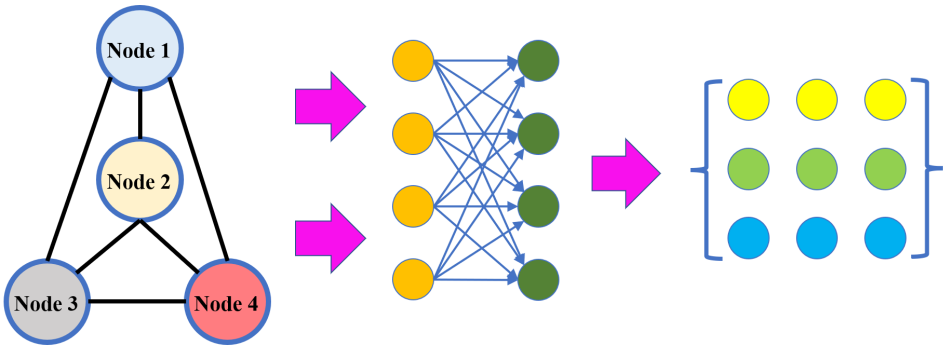



**FIGURE 3.** Illustration for main thought of GCN.

such frequency is set as four different values: 5 minutes, 10 minutes, 15 minutes and 20 minutes. Within one day, the four frequency values can generate four monitoring values: 288, 144, 96 and 72. There are totally 91 days in the dataset, and they constitute all the discrete data samples.

To evaluate contributions of the proposed DTFUN, three forecasting models that can be used for sequential modeling are selected as the baseline methods. The DTFUN will be compared with them with respect to performance indexes for assessment. The selected three approaches are named as: LSTM, GRU, and CNN, respectively. They are briefly described as follows:

- ***LSTM:*** It refers to long short-term memory (LSTM) model which is a typical sequential modeling method using the structure of neural computing.
- ***GRU:*** It refers to gated recurrent unit (GRU) model which is also a typical sequential modeling method using neural computing. It is actually a revised version of LSTM.
- ***CNN:*** It refers to convolution neural network (CNN) model which is a typical neural network structure. It integrates the convolution operations into the neural computing process.
- ***TGCN:*** It refers to the temporal graph convolution network (TGCN) model which is an improved GCN structure. It integrates the temporal modeling into the graph convolution operations [33].
- ***STGCN:*** It refers to the spatial-temporal graph convolution netwrok (STGCN) model which is also an improved GCN structure. It integrates the spatial-temporal information into the graph convolution operations [34].

For performance measurement, there are also two major metrics employed: MAE and RMSE. They are briefly described as follows:

- ***MAE:*** It refers to mean average error, and calculates the average error between real values and predicted values. The unit error is measured using absolute values. The MAE can be defined as the following formula:

$$MAE = \frac{1}{I}\sum_{i=1}^{I}|p_i - r_i| \tag{21}$$

- ***RMSE:*** It refers to rooted mean squared error, and uses a formula similar to Euclidean distance to measure error between real values and predicted values. The RMSE can be defined as the following formula:

$$RMSE = \sqrt{\frac{1}{I}\sum_{i=1}^{I}(p_i - r_i)^2} \tag{22}$$

Besides the two metrics, another evaluation metrics that combines the two metrics together is introduced for comparison. The specifically developed metric is defined as the sum of MAE value and RMSE value, and is named as SMR here. The SMR is calculated as follows:

$$SMR = \frac{1}{2}(MAE + RMSE) \tag{23}$$

For the dataset, it is divided into training part and testing part. The former is used to train the forecasting models, and the latter is used to evaluate the trained models. The proportion of training data is uniformly set to the value of 80%. The model does not directly have some hyperparameters, while the training process has the hyperparameters. The learning rate is set to 0.001 and 0.002 to construct different scenarios. During training process, the batch size is set to 32, and the epoch number is set to 10. The Adaptive momentum (Adam) algorithm is selected as the optimizer for training process. The internal parameters inside the Adam are set as their default values. As the initial dataset is within the format of continuous values, it is expected to take sampling operations for it. Four sampling frequencies are selected here: 5 minutes, 10 minutes, 15 minutes, and 20 minutes.

### B. RESULTS

Main experimental results (MAE and RMSE) are demonstrated in Table 3 and Table 4. The two tables have seven lines and nine columns. The first line lists main experimental setting information, and other lines list experimental results. The first column lists four experimental methods which contain the proposal and baselines, and the other eight columns list experimental results. Among, the second, fourth, sixth and eighth lines correspond to MAE results, and the third, fifth, seventh and ninth lines correspond to RMSE results. There are four groups of MAE and RMSE values in each table,

**TABLE 3.** Main experimental results (MAE and RMSE) with learning rate being set as 0.001.

| Methods | 5 minutes | | 10 minutes | | 15 minutes | | 20 minutes | |
|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| CNN | 23.21 | 34.82 | 23.73 | 37.26 | 24.86 | 38.28 | 25.15 | 41.25 |
| GRU | 22.29 | 33.21 | 23.31 | 35.66 | 23.84 | 37.91 | 23.67 | 36.69 |
| LSTM | 22.62 | 35.74 | 23.43 | 35.04 | 23.91 | 37.11 | 24.36 | 37.27 |
| TGCN | 20.86 | 32.51 | 21.44 | 32.23 | 22.38 | 34.16 | 22.53 | 33.89 |
| STGCN | 20.68 | 32.07 | 21.60 | 32.59 | 22.15 | 33.87 | 22.60 | 34.18 |
| DTFUN | 20.99 | 32.48 | 21.68 | 32.87 | 22.46 | 34.32 | 22.72 | 34.51 |

**TABLE 4.** Main experimental results (MAE and RMSE) with learning rate being set as 0.002.

| Methods | 5 minutes | | 10 minutes | | 15 minutes | | 20 minutes | |
|---|---|---|---|---|---|---|---|---|
| | MAE | RMSE | MAE | RMSE | MAE | RMSE | MAE | RMSE |
| CNN | 23.89 | 37.51 | 24.37 | 39.72 | 25.06 | 39.34 | 25.75 | 40.69 |
| GRU | 21.92 | 34.98 | 22.55 | 36.87 | 23.48 | 37.81 | 24.19 | 39.16 |
| LSTM | 23.09 | 37.34 | 23.54 | 37.19 | 24.26 | 38.87 | 24.61 | 38.59 |
| TGCN | 21.27 | 34.03 | 21.21 | 34.89 | 21.49 | 35.97 | 21.48 | 35.23 |
| STGCN | 21.08 | 33.57 | 21.12 | 34.64 | 21.26 | 35.53 | 21.56 | 35.19 |
| DTFUN | 21.05 | 33.47 | 21.38 | 35.28 | 21.36 | 35.66 | 21.79 | 35.86 |

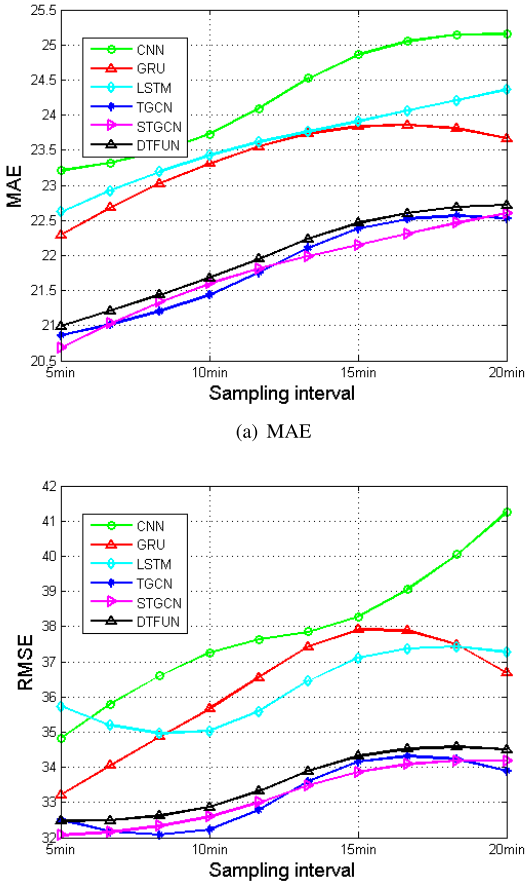
which correspond to four scenarios when sampling interval is set to 5 minutes, 10 minutes, 15 minutes and 20 minutes. It can be seen from the two tables that experimental results can be achieved better when learning rate is set to 0.001. And the five baseline methods have relatively fluctuating performance presentation under different parameter setting. Compared with three basic methods: CNN, LSTM, GRU, the DTFUN can always attain better experimental results. When it comes to two GCN-based methods: TGCN and STGCN, the DTFUN may not have significant performance advantage compared with them. Because they consider more fine-grained features when modeling traffic networks. Thus, we will discuss the time complexity in the following contents.

In order to achieve better visualization effect, main results in Table 3 and Table 4 are also demonstrated in the format of curve diagrams. The Fig. 4 and Fig. 5 are the curve diagrams to illustrate MAE and RMSE results. Each of them has two subfigures, corresponding to MAE results and RMSE results, separately. For each subfigure, its X-axis corresponds to sampling interval that ranges from 5 minutes to 20 minutes. For these methods, MAE values and RMSE values show an ascending tendency on the whole. And sometimes GRU and LSTM have some fluctuation in performance tendency. It can be clearly observed from these subfigures that the curves of DTFUN, TGCN and STGCN are always below the curves of other methods. This phenomenon can show better forecasting performance of DTFUN compared than three basic methods: CNN, LSTM and GRU. Because the MAE and RMSE are used to measure distance between predicted values and real values. Lower values of them denote better forecasting performance. For three GCN-based methods: DTFUN, TGCN and STGCN, they all have their own advantage situations when different parameters are set. Overall, the DTFUN may not show a remarkable performance advantage compared with other two.

Combining MAE and RMSE together, the metric SMR is utilized for assessment. Related results are illustrated in Table 5 and Fig. 6. The Table 5 has seven lines and nine columns. The first line lists some experimental setting conditions that contain two learning rate values, and the other six lines list experimental results of four methods. The first column lists six experimental methods, and other eight columns list experimental results. And the main results in Table 5 are also visualized in Fig. 6 for better visualization effect. It is composed of two subfigures which correspond to SMR results under learning rate of 0.001 and 0.002, respectively. The X-axis denotes sampling interval that ranges from 5 minutes to 20 minutes, and the Y-axis denotes SMR values. It can be seen from both Table 5 and Fig. 6 that the proposed DTFUN have proper performance in the experiments. Although DTFUN cannot have better performance compared with two temporal forecasting methods:

**TABLE 5.** SMR results with learning rate being set as 0.001 and 0.002.

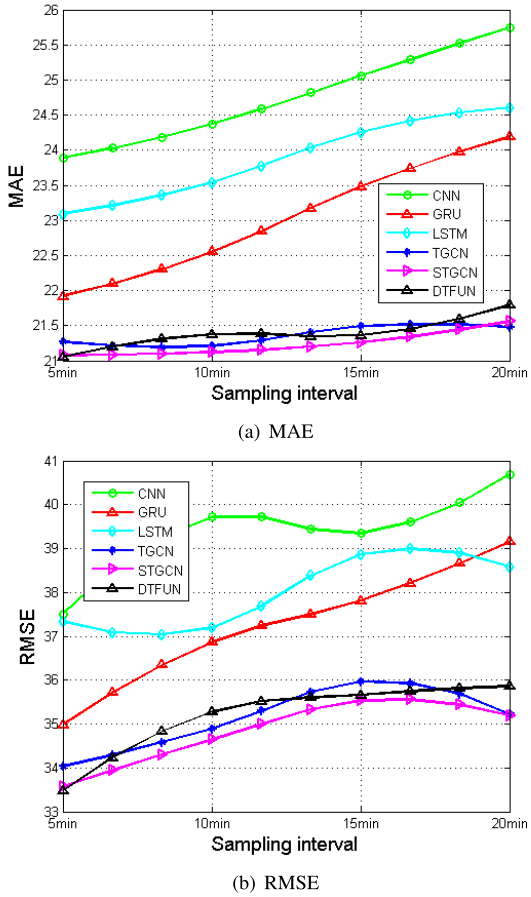
| Methods | Learning rate: 0.001 | | | | Learning rate: 0.002 | | | |
|---|---|---|---|---|---|---|---|---|
| | 5min | 10min | 15min | 20min | 5min | 10min | 15min | 20min |
| CNN | 29.02 | 30.50 | 31.57 | 33.20 | 30.70 | 32.05 | 32.20 | 33.22 |
| GRU | 27.75 | 29.49 | 30.88 | 30.18 | 28.45 | 29.71 | 30.65 | 31.68 |
| LSTM | 29.18 | 29.24 | 30.51 | 30.82 | 30.22 | 30.37 | 31.57 | 31.60 |
| TGCN | 26.69 | 26.84 | 28.27 | 28.21 | 27.65 | 28.05 | 28.73 | 28.36 |
| STGCN | 26.38 | 27.10 | 28.01 | 28.39 | 27.33 | 27.88 | 28.40 | 28.38 |
| DTFUN | 26.74 | 27.28 | 28.39 | 28.62 | 27.26 | 28.33 | 28.51 | 28.83 |




(a) MAE

(b) RMSE

**FIGURE 4.** Main results in terms of two metrics under learning rate of 0.001.



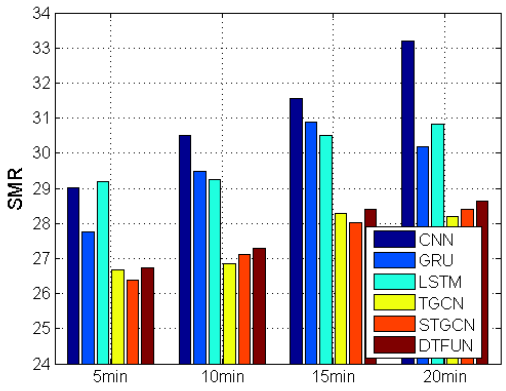

(a) MAE

(b) RMSE

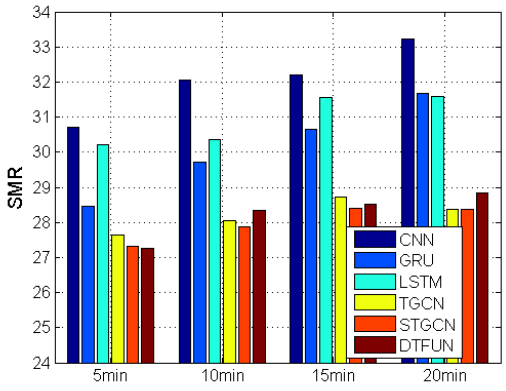
**FIGURE 5.** Main results in terms of two metrics under learning rate of 0.002.

TGCN and STGCN, it can perform better than general deep learning-based forecasting methods.

In addition to the forecasting performance, it is expected to testify the time complexity for experimental methods. It has been mentioned that $I$ denotes size of road nodes, $T$ denotes size of timestamps, $n$ denotes size of feature factors in graph convolution. Besides, the hidden size of both LSTM and GRU is denoted as $S_h$, the size of adjacency matrix in GCN is denoted as $S_A$, the epoch number of all experimental methods is denoted as $S_e$, the kernel size of convolution operations is denoted as $S_k$, channel size of convolution operations is denoted as $S_c$, and attention size inside STGCN model is denoted as $S_{atten}$. For CNN, its time complexity is represented as $\mathcal{O}(I \cdot S_k \cdot S_c \cdot S_e)$. For LSTM and GRU, their time complexity is represented as $\mathcal{O}[(I + S_h) \times S_h \cdot S_e]$. For DTFUN, its time complexity is represented as $\mathcal{O}[I \cdot T \cdot S_A \cdot S_e]$. For TGCN, it has two main parts: GCN part and temporal modeling part. Its time complexity is represented as $O[(I+S_h)\times S_h \cdot T \cdot S_A \cdot S_e]$. For STGCN, it has another attention computation, and its




(a) 0.001




(b) 0.002

**FIGURE 6.** SMR results under two different learning rate values: 0.001 and 0.002.

**TABLE 6.** Time complexity of experimental methods.

| Methods | Complexity |
|---|---|
| CNN | $\mathcal{O}(I \cdot S_k \cdot S_c \cdot S_e)$ |
| LSTM | $\mathcal{O}[(I + S_h) \cdot S_h \cdot S_e]$ |
| GRU | $\mathcal{O}[(I + S_h) \cdot S_h \cdot S_e]$ |
| DTFUN | $\mathcal{O}[I \cdot T \cdot S_A \cdot S_e]$ |
| TGCN | $O[(I + S_h) \cdot S_h \cdot T \cdot S_A \cdot S_e]$ |
| STGCN | $O[(I + S_h) \cdot S_h \cdot T \cdot S_{atten} \cdot S_A \cdot S_e]$ |

complexity can be represented as $O[(I+S_h)\times S_h \cdot T \cdot S_{atten} \cdot S_A \cdot S_e]$. The time complexity comparison is further demonstrated in TABLE 6.

In the experiments, we use three evaluation metrics to assess forecasting performance of the proposed DTFUN. It can be seen from the results that DTFUN can perform better than general deep learning-based forecasting methods. But when it comes to recent graph convolution-based methods, the DTFUN may not show obvious performance superiority. Thus, we make further discussion for time complexity. It can be seen from TABLE 6 that DTFUN has less time complexity compared with other two graph convolution-based methods. The DTFUN has the complexity level of $\mathcal{O}[I \cdot T \cdot S_A \cdot S_e]$, the TGCN has the complexity level of $O[(I + S_h) \cdot S_h \cdot T \cdot S_A \cdot S_e]$, and the STGCN has the complexity level of $O[(I + S_h) \cdot S_h \cdot T \cdot S_{atten} \cdot S_A \cdot S_e]$. The latter two methods have large time complexity than the DTFUN. Considering that topic of this paper focuses on a fast traffic flow forecasting method. This requires that technical methods have less time complexity and better practice. From this point, the DTFUN still has some overall advantage compared with baseline methods.

## V. CONCLUSION

The graph-level forecasting is a promising means for traffic flow in modern urban road networks. This can be expected to promote forecasting effect of traditional methods. This paper discusses feasibility of graph convolution theory, and introduces the a graph deep learning method named GCN to construct a graph-level forecasting method for traffic flow. The proposed method is named as DTFUN for short. In experiments, it is compared with three other methods that do not use graph learning for assessment. A real standard dataset for traffic flow is selected as the simulation scenario, and obtained results can well verify the proposal's performance presentation. Two aspects of views can be concluded from the experiments.

- The GCN can perform better than other deep learning-based general forecasting methods, because its thought can well fit structure of road networks.
- The GCN can reduce forecasting error about 5%-10% compared with several typical deep learning-based forecasting methods.
- The GCN can serve as a baseline for traffic flow forecasting tasks, and it can be extended for realistic engineering applications.

In future works, the authors are going to pay attention to a new kind of network entity named as social vehicular networks. Under such media, more personalized service such as traffic path recommendation [40] can be carried out for users, according to traffic flow forecasting results.

## REFERENCES

[1] L. Zhao, Z. Bi, A. Hawbani, K. Yu, Y. Zhang, and M. Guizani, "ELITE: An intelligent digital twin-based hierarchical routing scheme for softwarized vehicular networks," *IEEE Trans. Mobile Comput.*, vol. 22, no. 9, pp. 5231–5247, Sep. 2022.

[2] S. Xia, Z. Yao, G. Wu, and Y. Li, "Distributed offloading for cooperative intelligent transportation under heterogeneous networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16701–16714, Sep. 2022.

[3] Z. Guo, K. Yu, N. Kumar, W. Wei, S. Mumtaz, and M. Guizani, "Deep-distributed-learning-based POI recommendation under mobile-edge networks," *IEEE Internet Things J.*, vol. 10, no. 1, pp. 303–317, Jan. 2023.

[4] Z. Zhou, X. Dong, Z. Li, K. Yu, C. Ding, and Y. Yang, "Spatio-temporal feature encoding for traffic accident detection in VANET environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 10, pp. 19772–19781, Oct. 2022.

[5] Z. Guo, K. Yu, Z. Lv, K. R. Choo, P. Shi, and J. J. P. C. Rodrigues, "Deep federated learning enhanced secure POI microservices for cyber-physical systems," *IEEE Wireless Commun.*, vol. 29, no. 2, pp. 22–29, Apr. 2022.

[6] Y. Li, H. Ma, L. Wang, S. Mao, and G. Wang, "Optimized content caching and user association for edge computing in densely deployed heterogeneous networks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 6, pp. 2130–2142, Jun. 2022.

[7] B. Zhu, K. Chi, J. Liu, K. Yu, and S. Mumtaz, "Efficient offloading for minimizing task computation delay of NOMA-based multiaccess edge computing," *IEEE Trans. Commun.*, vol. 70, no. 5, pp. 3186–3203, May 2022.

[8] D. Peng, D. He, Y. Li, and Z. Wang, "Integrating terrestrial and satellite multibeam systems toward 6G: Techniques and challenges for interference mitigation," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 24–31, Feb. 2022.

[9] Y. He, L. Nie, T. Guo, K. Kaur, M. M. Hassan, and K. Yu, "A NOMA-enabled framework for relay deployment and network optimization in double-layer airborne access VANETs," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 22452–22466, Nov. 2022.

[10] Y. Sun, K. Yu, A. K. Bashir, and X. Liao, "Bl-IEA: A bit-level image encryption algorithm for cognitive services in intelligent transportation systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 1, pp. 1062–1074, Jan. 2023.

[11] E. Macioszek, "Changes in values of traffic volume-case study based on general traffic measurements in Opolskie Voivodeship (Poland)," in *Proc. Directions Develop. Transp. Netw. Traffic Eng., 15th Sci. Tech. Conf. Transp. Syst. Theory Pract.* Katowice, Poland: Springer, Sep. 2019, pp. 66–76.

[12] E. Macioszek and A. Kurek, "Extracting road traffic volume in the city before and during COVID-19 through video remote sensing," *Remote Sens.*, vol. 13, no. 12, p. 2329, 2021.

[13] Q. Zhang, Z. Guo, Y. Zhu, P. Vijayakumar, A. Castiglione, and B. B. Gupta, "A deep learning-based fast fake news detection model for cyber-physical social services," *Pattern Recognit. Lett.*, vol. 168, pp. 31–38, 2023.

[14] L. Huang, R. Nan, K. Chi, Q. Hua, K. Yu, N. Kumar, and M. Guizani, "Throughput guarantees for multi-cell wireless powered communication networks with non-orthogonal multiple access," *IEEE Trans. Veh. Technol.*, vol. 71, no. 11, pp. 12104–12116, Nov. 2022.

[15] 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 Trans. Intell. Transp. Syst.*, vol. 23, no. 2, pp. 1009–1018, Feb. 2022.

[16] C. Furtlehner, J.-M. Lasgouttes, A. Attanasi, M. Pezzulla, and G. Gentile, "Short-term forecasting of urban traffic using spatio-temporal Markov field," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 10858–10867, Aug. 2022.

[17] Z. Guo, K. Yu, K. Konstantin, S. Mumtaz, W. Wei, P. Shi, and J. J. P. C. Rodrigues, "Deep collaborative intelligence-driven traffic forecasting in green Internet of Vehicles," *IEEE Trans. Green Commun. Netw.*, vol. 7, no. 2, pp. 1023–1035, Jun. 2023.

[18] Q. Zhang, K. Yu, Z. Guo, S. Garg, J. J. P. C. Rodrigues, M. M. Hassan, and M. Guizani, "Graph neural network-driven traffic forecasting for the connected Internet of Vehicles," *IEEE Trans. Netw. Sci. Eng.*, vol. 9, no. 5, pp. 3015–3027, Sep. 2022.

[19] X. Ta, Z. Liu, X. Hu, L. Yu, L. Sun, and B. Du, "Adaptive spatio-temporal graph neural network for traffic forecasting," *Knowl.-Based Syst.*, vol. 242, Apr. 2022, Art. no. 108199.

[20] A. Khaled, A. M. T. Elsir, and Y. Shen, "TFGAN: Traffic forecasting using generative adversarial network with multi-graph convolutional network," *Knowl.-Based Syst.*, vol. 249, Aug. 2022, Art. no. 108990.

[21] S. Zhang, Y. Guo, P. Zhao, C. Zheng, and X. Chen, "A graph-based temporal attention framework for multi-sensor traffic flow forecasting," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 7743–7758, Jul. 2022.

[22] R. He, Y. Liu, Y. Xiao, X. Lu, and S. Zhang, "Deep spatio-temporal 3D Densenet with multiscale ConvLSTM-Resnet network for citywide traffic flow forecasting," *Knowl.-Based Syst.*, vol. 250, Aug. 2022, Art. no. 109054.

[23] T. Zhang and G. Guo, "Graph attention LSTM: A spatiotemporal approach for traffic flow forecasting," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, no. 2, pp. 190–196, Mar. 2022.

[24] Y. Wang, C. Jing, S. Xu, and T. Guo, "Attention based spatiotemporal graph attention networks for traffic flow forecasting," *Inf. Sci.*, vol. 607, pp. 869–883, Aug. 2022.

[25] C. Zhang, L. Cui, S. Yu, and J. J. Q. Yu, "A communication-efficient federated learning scheme for IoT-based traffic forecasting," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11918–11931, Jul. 2022.

[26] Y. Wang, S. Fang, C. Zhang, S. Xiang, and C. Pan, "TVGCN: Time-variant graph convolutional network for traffic forecasting," *Neurocomputing*, vol. 471, pp. 118–129, Jan. 2022.

[27] Y. Fang, S. Ergüt, and P. Patras, "SDGNet: A handover-aware spatiotemporal graph neural network for mobile traffic forecasting," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 582–586, Mar. 2022.

[28] J. Zhao, Z. Liu, Q. Sun, Q. Li, X. Jia, and R. Zhang, "Attention-based dynamic spatial–temporal graph convolutional networks for traffic speed forecasting," *Exp. Syst. Appl.*, vol. 204, Oct. 2022, Art. no. 117511.

[29] F. Carpio, W. Bziuk, and A. Jukan, "Scaling migrations and replications of virtual network functions based on network traffic forecasting," *Comput. Netw.*, vol. 203, Feb. 2022, Art. no. 108582.

[30] J. Liu, N. Wu, Y. Qiao, and Z. Li, "Short-term traffic flow forecasting using ensemble approach based on deep belief networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 404–417, Jan. 2022.

[31] S. Reza, M. C. Ferreira, J. J. M. Machado, and J. M. R. S. Tavares, "A multi-head attention-based transformer model for traffic flow forecasting with a comparative analysis to recurrent neural networks," *Exp. Syst. Appl.*, vol. 202, Sep. 2022, Art. no. 117275.

[32] J.-H. Duan, W. Li, X. Zhang, and S. Lu, "Forecasting fine-grained city-scale cellular traffic with sparse crowdsourced measurements," *Comput. Netw.*, vol. 214, Sep. 2022, Art. no. 109156.

[33] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3848–3858, Sep. 2020.

[34] X. Wang, Y. Ma, Y. Wang, W. Jin, X. Wang, J. Tang, C. Jia, and J. Yu, "Traffic flow prediction via spatial temporal graph neural network," in *Proc. Web Conf.*, Taipei, Taiwan, Apr. 2020, pp. 1082–1092.

[35] H. Wang, R. Zhang, X. Cheng, and L. Yang, "Hierarchical traffic flow prediction based on spatial–temporal graph convolutional network," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 16137–16147, Sep. 2022.

[36] Q. Lai, J. Tian, W. Wang, and X. Hu, "Spatial–temporal attention graph convolution network on edge cloud for traffic flow prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 24, no. 4, pp. 4565–4576, Apr. 2023.

[37] T. Qi, L. Chen, G. Li, Y. Li, and C. Wang, "FedAGCN: A traffic flow prediction framework based on federated learning and asynchronous graph convolutional network," *Appl. Soft Comput.*, vol. 138, May 2023, Art. no. 110175.

[38] Y. Zheng, W. Li, W. Zheng, C. Dong, S. Wang, and Q. Chen, "Lane-level heterogeneous traffic flow prediction: A spatiotemporal attention-based encoder–decoder model," *IEEE Intell. Transp. Syst. Mag.*, vol. 15, no. 3, pp. 51–67, May 2023.

[39] Y. Duan, N. Chen, S. Shen, P. Zhang, Y. Qu, and S. Yu, "FDSA-STG: Fully dynamic self-attention spatio-temporal graph networks for intelligent traffic flow prediction," *IEEE Trans. Veh. Technol.*, vol. 71, no. 9, pp. 9250–9260, Sep. 2022.

[40] Z. Guo, C. Tang, W. Niu, Y. Fu, T. Wu, H. Xia, and H. Tang, "Fine-grained recommendation mechanism to curb astroturfing in crowdsourcing systems," *IEEE Access*, vol. 5, pp. 15529–15541, 2017.

**DONGFANG YANG** received the B.Sc. degree in computer science and technology and the M.Sc. degree in computer application technology from Zhengzhou University, in 2005 and 2012, respectively. She is currently an Associate Professor with the School of Information Engineering, Zhengzhou Shengda University, China. Her research interests include edge intelligence, intelligent transportation systems, and big data.

**LIPING LV** received the B.Sc. degree in industrial automation from the Qingdao Institute of Architecture and Engineering, in 2001, and the M.Sc. degree in control theory and control engineering from the Qingdao University of Science and Technology, in 2005. She is currently a Professor with the School of Information Engineering, Zhengzhou Shengda University, China. Her research interests include information processing, edge intelligence, wireless sensor networks, and artificial intelligence.