

## Research Article

# MDGCN: Multiple Graph Convolutional Network Based on the Differential Calculation for Passenger Flow Forecasting in Urban Rail Transit

Chenxi Wang, Huizhen Zhang , Shuilin Yao, Wenlong Yu, and Ming Ye

College of Computer Science and Technology, Huaqiao University, Xiamen, China

Correspondence should be addressed to Huizhen Zhang; zhanghz@hqu.edu.cn

Received 23 July 2021; Revised 24 October 2021; Accepted 5 November 2021; Published 1 December 2021

Academic Editor: Geqi Qi

Copyright © 2021 Chenxi Wang 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.

Passenger flow forecasting plays an important role in urban rail transit (URT) management. However, complex spatial and temporal correlations make this task extremely challenging. Previous work has been done by capturing spatiotemporal correlations of historical data. However, the spatiotemporal relationship between stations not only is limited to geospatial adjacency, but also lacks different perspectives of station correlation analysis. To fully capture the spatiotemporal correlations, we propose a deep learning model based on graph convolutional neural networks called MDGCN. Firstly, we identify the heterogeneity of stations under two spaces by the Multi-graph convolutional layer. Secondly, we designed the Diff-graph convolutional layer to identify the changing trend of heterogeneous features and used the attention mechanism unit with the LSTM unit to achieve adaptive fusion of multiple features and modeling of temporal correlation. We evaluate this model on real datasets. Compared to the best baselines, the root-mean-square errors of MDGCN are improved by 1%–15% for different prediction intervals.

## 1. Introduction

With the expansion of urban traffic congestion, URT has become one of the important solutions to alleviate the congestion problem. As an important research topic in URT, short time passenger flow forecasting can help decision-makers to make timely emergency plans and enhance security forces [1], as well as providing an important reference basis for the optimization of bus line networks.

Accurate, real-time passenger flow forecasting in URT is a challenging task. This is because the accuracy of forecasting can be affected by various aspects, such as the URT network's development, its topology, and spatiotemporal correlation. The short-term prediction models of URT passenger flow range from the initial mathematical-statistical models such as Autoregressive Integrated Moving Average (ARIMA) [2] to traditional machine learning models such as Support Vector Machines (SVM) [3] to various deep learning models. However, mathematical-statistical models cannot mine the nonlinear features of traffic flow data, and traditional machine learning models rely on feature extraction, so

their adaptability to data is not stable. Compared with traditional mathematical-statistical models and machine learning models, deep learning models perform better in the extraction of spatiotemporal features and are mainly classified into three categories: CNN-based, RNN-based, and GCN-based.

In the earliest stage, many scholars treated traffic data as image pixels and used a convolutional neural network (CNN) for local spatiotemporal feature extraction. For example, Luo et al. [4] modeled the traffic flow as a two-dimensional spatiotemporal matrix to describe the images of spatiotemporal relationships of traffic flow and used CNN to predict the traffic speed. Similarly, based on converting the passenger flow data into images, Zhang et al. [5] introduced the residuals into CNN. However, applying CNN to the passenger flow prediction cannot accurately capture the spatiotemporal correlation. Regarding this problem, Zhang et al. [6] proposed a multitask learning prediction model called MTL-TCNN that considered the relevance of multiple regions for the prediction problem, but CNN is only suitable for Euclidean space, and traffic data is typical non-Euclidean

data. Recurrent neural network (RNN) and its variant models have good performance in handling time-series data because of their self-circulation mechanism. For example, Fu et al. [7] used gate recurrent unit (GRU) for traffic flow prediction for the first time. Ma et al. [8] and Yang et al. [9] used long-short term memory (LSTM) to capture the trend of long and short time series. Abduljabbar et al. [10] extended the one-way LSTM into a two-way LSTM; thus, the accuracy was further improved by training the data twice by forward and backward. However, RNN and CNN cannot reasonably extract the correlation of spatial dimensions and are not suitable for traffic flow data processing.

Researchers gradually realize the advantages of GCN in dealing with non-Euclidean data such as traffic flow. Many scholars tend to apply GCN to traffic flow prediction with other models that perform well in capturing temporal correlations. Some studies [11–13] integrated GCN- and RNN-based models to predict. For example, Zhao et al. [14] and Lv et al. [15] combined GCN with GRU; Geng et al. [16] combined GCN with RNN; Zhang et al. [1], Chai et al. [17], and He et al. [18] combined GCN with LSTM to make up for the shortcoming of GCN in capturing temporal dependence poorly. Zhang et al. [19] proposed a model including attention mechanism, GCN, and sequence-to-sequence for multistep prediction. Besides, some studies tried to propose novel models, such as Wu et al. [20], who proposed an adaptive adjacency matrix to change the structure of a fixed graph and used extended convolution to capture temporal correlation. Park et al. [21] used the transformer model [22] and the self-attention mechanism with encoder-decoder architecture. Hao et al. [23] constructed a sequence-to-sequence architecture with attentional mechanisms. Also, a hybrid prediction model [24] based on kernel ridge regression (KRR) and Gaussian process regression (GPR) is proposed to predict the short-term passenger flow. However, these studies only consider the topological structure of the URT network. More correlations between stations should be explored. Effectively exploring correlations among stations could further improve the prediction performance. For example, Geng et al. [16], Chai et al. [17], Wang et al. [25], and Wang et al. [26] reconstructed the relationship graph from different perspectives. However, they ignored the correlation hidden in historical data and the changing trend which could help capture the movement patterns of passenger flow.

The above research indicates that there are still two problems to be solved in the current research on URT passenger flow prediction: the correlation between stations and the extraction of changing trends. To be exact, the relationship between stations is not limited to the adjacency relationship of geographical space; there is a lack of station correlation analysis from different perspectives. And it is necessary to construct an appropriate model to mine the changing trends.

In this paper, we propose MDGCN (Multi-Graph Differential Convolutional Network). It can mine the correlations between stations in heterogeneous space for URT passenger flow prediction. The contributions of this paper are as follows:

- (i) The correlation of boarding and alighting passenger flows between stations that are physically distant from each other is also important for passenger flow prediction. Therefore, to more accurately represent the connectivity between stations, we reconstruct the connectivity of stations by historical passenger flow data and station geographic information.
- (ii) We design a layer of Multi-GCN that can jointly mine the correlation between stations in physical space and nonphysical space to gain the heterogeneous spatial correlation between stations.
- (iii) We introduce the concept of difference and construct the difference feature extraction layer called Diff-GCN to extract the changing trends of heterogeneous spatial features.
- (iv) We conducted experiments on two datasets. The experimental results show that the prediction error of the proposed model is reduced by 1%–15% compared with the optimal baseline.

The arrangement of the paper is shown below. Section 2 introduces the basic problem definition of passenger flow forecasting. Section 3 presents the model framework. Section 4 discusses the experimental results. Conclusions are made in Section 5.

## 2. Preliminary

In this section, we define some key concepts and give the problem definition of the studied content.

*Definition 1 (urban rail transit network).* We define the URT network using the graph as  $G(V, E, A)$ , where the set of stations is used as nodes  $V = \{v_1, v_2, \dots, v_n\}$  of  $G$ , and  $n$  is the number of stations. Figure 1 illustrates sample lines related to the URT network. The connection relationships between stations are used as edges  $E = \{e_k: (v_i, v_j) \in V \times V\}$  of  $G$ . We use the matrix  $A \in R^{n \times n}$  to record the adjacent relationships between stations, and the value  $\delta_{i,j}$  represents the strength of the association between stations. So, we reconstruct the spatial relationships of the URT network from two perspectives.

*Definition 2 (node characteristics).* We take the passenger flow of a station as the characteristics of a network node; i.e., given a station  $V_i$  and a period  $t$ , the passenger flow of the station at that period is denoted as  $X_t^{V_i}$ , and the passenger flow of all stations in the past  $m$  periods is denoted as  $X = \{X_t, X_{t-1}, X_{t-2}, \dots, X_{t-m+1}\}$ ; then the temporal characteristics of the set  $V$  of stations in the past  $m$  periods can be written as  $X \in R^{n \times m}$ .

*Problem. Definition.* This problem can be defined as learning a mapping function to predict the passenger flow data at the next moment given the URT network  $G(V, E, A)$  with the historical passenger flow data  $X$ . The process can be referred to as

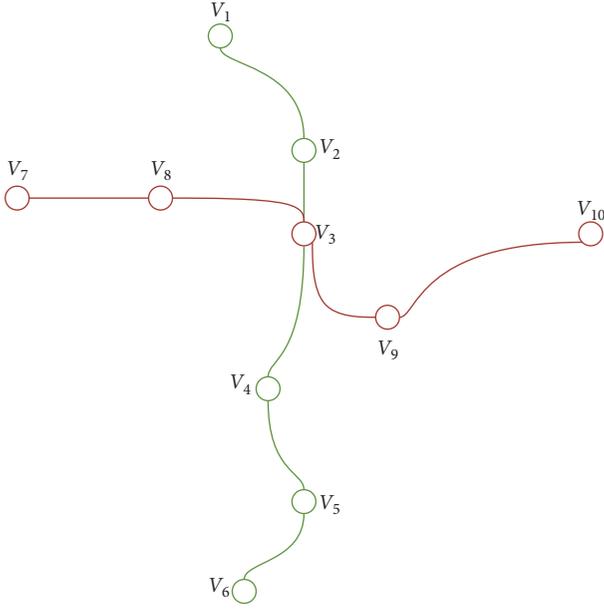


FIGURE 1: A sample of an urban rail transit network.

$$X_{t+1} = f(G, X). \quad (1)$$

### 3. Study Methodology

This section introduces the framework of our model, research methods, and pseudocode.

**3.1. Modeling Framework.** This model (Figure 2) can be divided into four main parts: the reconstruction of the relationship of station, the layer of Multi-GCN, the layer of Diff-GCN, and the layer of output.

- (i) The first part is based on the relationship between the physical space and the nonphysical space of stations and combines the historical passenger flow data as the input of the subsequent model.

- (ii) The second part is the Multi-GCN (Multiple-Graph Convolutional Network) layer, which can jointly model the correlation between stations in both physical space and nonphysical space; i.e., the Multi-GCN layer is constructed to capture the heterogeneous spatial correlation between stations.
- (iii) The third part is a Diff-GCN (Differential Graph Convolutional Network) layer, a differential feature extraction module, to extract the changing trends of heterogeneous spatial features. Besides, the attention mechanism is used to adaptively fuse the features of the Multi-GCN layer and the Diff-GCN layer.
- (iv) The fourth part is the LSTM unit for global temporal correlation extraction and the fully connected network layer for prediction result output.

**3.1.1. Reconstruction of the Correlation of Stations.** The key for a neural network being able to be trained is in the input data. And the input to a GCN consists of two parts: the adjacency matrix and the node features. Whether the adjacency matrix correctly encodes the relationships of the network nodes is related to the performance of the model. Therefore, we interpret the URT network from two perspectives. Stations in close geographical proximity may have similar passenger flow patterns. Given this, we construct the topology of the physical space  $G_D(V, E_D, A_D)$  based on the physical adjacency and distances between station  $i$  and station  $j$ .

Firstly, we obtain the adjacency matrix  $A$  (in (2)) between stations using the physical adjacency relationship of stations. Then, we build the distance matrix  $D$  (in (3)) based on the inverse of the distance between stations. Finally, we reconstruct the topology matrix  $A_D$  using the Hadamard product formula [1] (in (4)).

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & a_{ij} & \vdots \\ a_{n1} & \cdots & \cdots & \cdots & a_{nn} \end{pmatrix}, \quad a_{ij} = \begin{cases} 1, & \text{if } i \text{ and } j \text{ is adjacent,} \\ 0, & \text{if } i \text{ and } j \text{ is not adjacent,} \end{cases} \quad (2)$$

$$D = \begin{pmatrix} d_{11} & d_{12} & d_{13} & \cdots & d_{1n} \\ d_{21} & d_{22} & d_{23} & \cdots & d_{2n} \\ d_{31} & d_{32} & d_{33} & \cdots & d_{3n} \\ \vdots & \vdots & \vdots & d_{ij} & \vdots \\ d_{n1} & d_{n2} & \cdots & \cdots & d_{nn} \end{pmatrix}, \quad d_{ij} = \begin{cases} \frac{1}{\text{distance}_{ij}}, & i \neq j, \\ 0, & i = j, \end{cases} \quad (3)$$

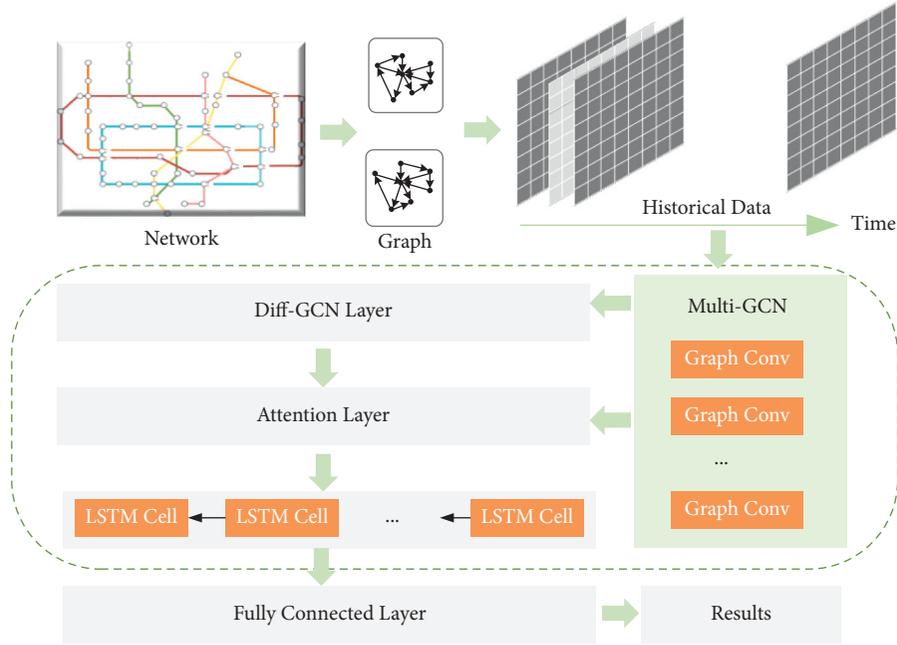


FIGURE 2: The architecture of MDGCN.

$$A_D = A \circ D. \quad (4)$$

However, the flow pattern could make two stations with no spatial adjacency have a certain association, and the strength of the association is determined by the passenger flow. This flow pattern truly reflects the trend of passenger flow movement between stations. Therefore, we describe the passenger flow movement pattern and movement intensity between stations based on the data of historical passenger flow (in (5)) to construct the associated graph  $G_C(V, E_C, A_C)$ . Denote  $p_{ij}$  as the passenger flow from station  $i$  to  $j$ .

$$A_C = \begin{pmatrix} p_{11} & p_{12} & p_{13} & \cdots & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdots & p_{2n} \\ p_{31} & p_{32} & p_{33} & \cdots & p_{3n} \\ \vdots & \vdots & \vdots & p_{ij} & \vdots \\ p_{n1} & p_{n2} & p_{n3} & \cdots & p_{nn} \end{pmatrix}. \quad (5)$$

**3.1.2. Multi-GCN Network.** Two different network topologies are obtained by spatial relationship reconstruction of stations. We propose the layer of Multi-GCN to extract heterogeneous spatial correlation by GCN. The propagation rule of GCN can be expressed as

$$H_{(l+1)} = \delta\left(\widehat{D}^{-(1/2)} \widehat{A} \widehat{D}^{-(1/2)} H_{(l)} W_{(l)}\right), \quad (6)$$

where  $H_{(l)}$  is the input of  $l$ -layer and its original value is  $X$ .  $\widehat{A} = \text{Normalization}(A + I)$  is the adjacent of a graph with added self-connections,  $\widehat{D} = \sum_j \widehat{A}_{ij}$  is degree matrix,  $\delta$  represents the sigmoid function for a nonlinear model, and  $W_{(l)}$  is a weight matrix of the current layer.

In particular, GCN has a powerful ability to model higher-order neighborhood interactions by stacking multiple layers. As shown in Figure 3, taking the red site as an example, the first layer of GCN can capture the influence of two neighboring stations of the red one. As the number of network layers deepens, the correlation of all stations can be captured completely. The spatial features in the URT network, i.e., the topology  $A_D$  based on the physical space and the station association matrix  $A_C$  in the passenger space, will be extracted by the  $N$ -layer GCN. So we input them into the layer of Multi-GCN to obtain the outputs  $Z_D$  (in (7)) and  $Z_C$  (in (8)) of the multigraph convolutional layers, respectively.

$$\begin{aligned} Z_D &= Z_{D(N)} \\ &= \delta\left[\widehat{D}^{-(1/2)} \widehat{A}_D \widehat{D}^{-(1/2)} Z_{D(N-1)} W\right], \end{aligned} \quad (7)$$

$$\begin{aligned} Z_C &= Z_{C(N)} \\ &= \delta\left[\widehat{D}^{-(1/2)} \widehat{A}_C \widehat{D}^{-(1/2)} Z_{C(N-1)} W\right]. \end{aligned} \quad (8)$$

**3.1.3. Diff-GCN Network.** The difference reflects the variation between two discrete quantities. We introduce the concept of differencing to construct the layer of Diff-GCN to extract the tendency of heterogeneous spatial features. As shown in Figure 4, the passenger flow data  $X_t, X_{t-1}$  based on time  $t$  and  $t-1$ , and the adjacent matrix  $A$  are put into Multi-GCN, respectively, and the respective feature matrices  $Z_{(N,t)}$  and  $Z_{(N,t-1)}$  are obtained after Multi-GCN, the difference is calculated, and the final difference factor  $\Delta Z$  (in (9)) is obtained by using GCN for spatial feature extraction.

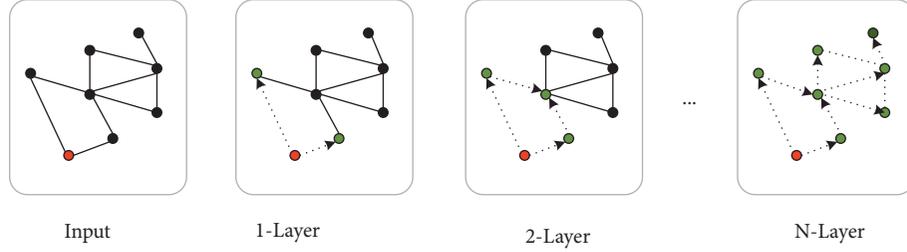


FIGURE 3: The process of capturing the spatial correlation of passenger flows using GCNs (taking red station as an example).

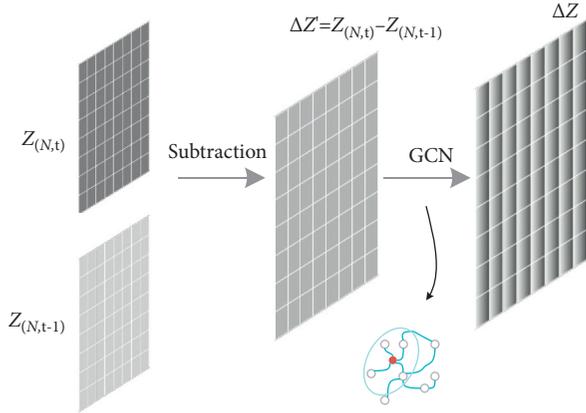


FIGURE 4: The architecture of Diff-GCN.

Here, we obtain two factors  $\Delta Z_D$  (in (10)) and  $\Delta Z_C$  (in (11)) based on two spatial relationships.

$$\Delta Z = \delta \left[ \hat{D}^{-(1/2)} \hat{A} \hat{D}^{-(1/2)} (Z_{(N,t)} - Z_{(N,t-1)}) W \right], \quad (9)$$

$$\begin{aligned} \Delta Z_D &= \Delta Z_{D(N)} \\ &= \delta \left[ \hat{D}^{-(1/2)} \hat{A}_D \hat{D}^{-(1/2)} (Z_{D(N,t)} - Z_{D(N,t-1)}) W \right], \end{aligned} \quad (10)$$

$$\begin{aligned} \Delta Z_C &= \Delta Z_{C(N)} \\ &= \delta \left[ \hat{D}^{-(1/2)} \hat{A}_C \hat{D}^{-(1/2)} (Z_{C(N,t)} - Z_{C(N,t-1)}) W \right]. \end{aligned} \quad (11)$$

**3.1.4. Attention.** For  $Z_D$  and  $Z_C$  obtained from the layer of Multi-GCN and  $\Delta Z_D$  and  $\Delta Z_C$  obtained from the layer of Diff-GCN, we use the attention mechanism to learn the importance  $\alpha_1, \alpha_2, \alpha_3, \alpha_4$  of these four features as follows (in (12)). Firstly, we calculate the hidden state vector scoring  $S_1$  of  $Z_D$  using a nonlinear transformation and one shared attention vector  $Q$  (in (13)). Here,  $W$  is the weight matrix and  $b$  is the bias vector. Similarly, the embedding  $Z_C, \Delta Z_D, \Delta Z_C$  do the same operation to get  $S_2, S_3, S_4$ . Secondly, we perform a normalization operation (in (14)) to obtain the coefficient ( $\alpha_i$ ) of the individual embedding vector. Finally, the linear operation is used to combine the four components to obtain  $Z$  (in (15)).

$$(\alpha_1, \alpha_2, \alpha_3, \alpha_4) = \text{Attention}(Z_D, Z_C, \Delta Z_D, \Delta Z_C), \quad (12)$$

$$S_1 = Q^T \cdot \tanh(W \cdot (Z_D)^T + b), \quad (13)$$

$$\begin{aligned} \alpha_i &= \text{softmax}(S_i) \\ &= \exp(S_i) / \sum_{i=1}^4 \exp(S_i), \end{aligned} \quad (14)$$

$$Z = \alpha_1 \cdot Z_D + \alpha_2 \cdot Z_C + \alpha_3 \cdot \Delta Z_D + \alpha_4 \cdot \Delta Z_C. \quad (15)$$

**3.1.5. Prediction.** After obtaining  $Z$  by attention, we use LSTM that is shown in Figure 5 to capture temporal features. LSTM is one of the most common neural network structures in time series forecasting problems. It uses the concept of gates to control the selective passage of information. As can be seen in (16) and Figure 5, it has three gates: Forget Gate, Input Gate, and Output Gate. Forget Gate filters the input information. Input Gate decides what new memory is stored. And Output Gate gets the processed results. The processing of each gate can be seen in (16).

As can be seen in Figure 5, after  $\{Z|Z_1, Z_2, \dots, Z_t\}$  is inputted into LSTM, the result  $\{h|h_1, h_2, \dots, h_t\}$  is gained. Then we obtain the predicted output  $\{X_{t+1}|X_{t+1}^V, X_{t+1}^V, \dots, X_{t+1}^V\}$  via the layer of fully connected with  $n$  neurons which means the number of stations.

$$\begin{cases} \text{Forget Gate: } f_t = \sigma(W_f \cdot [h_{t-1}, z_t] + b_f), \\ \text{Input Gate: } i_t = \sigma(W_i \cdot [h_{t-1}, z_t] + b_i), \\ \tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, z_t] + b_c), \\ C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t, \\ \text{Output Gate: } o_t = \sigma(W_o \cdot [h_{t-1}, z_t] + b_o), \\ h_t = o_t \cdot \tanh(C_t). \end{cases} \quad (16)$$

**3.2. Pseudocode.** MDGCN pseudocode is shown in Algorithm 1. Besides, to evaluate the complexity of this deep learning algorithm, we use two indicators, FLOPs and Params. The time complexity is determined by the number of operations of the model, i.e., FLOPs (0.0126 G), and the space complexity is determined by the number of parameters, i.e., Params (2, 147, 895).

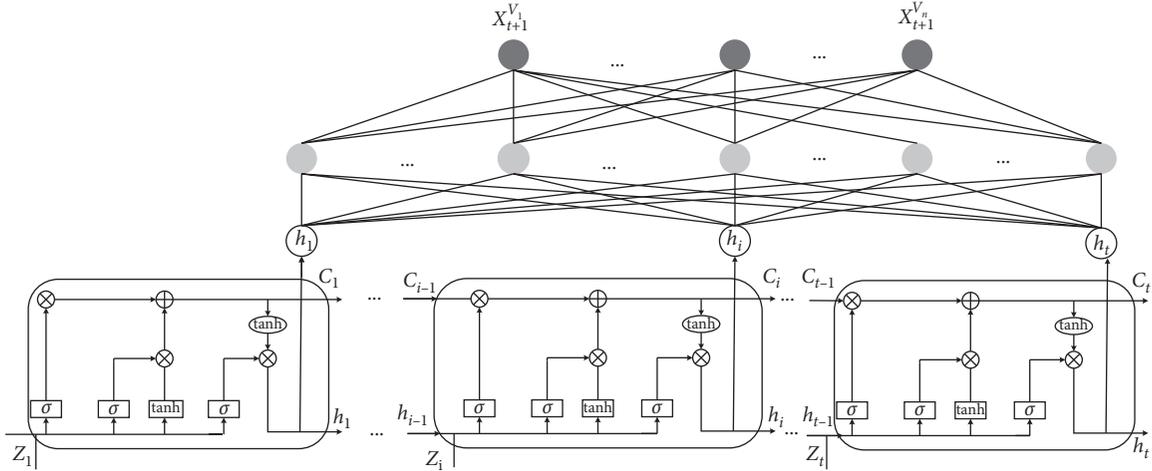


FIGURE 5: The architecture of prediction.

Input: Historical passenger flow  $\{X|X_1, X_2, \dots, X_t\}$  (described in Section 4.2.);  $G_C, G_D$  (described in Section 3.1.1).  
Output: Passenger flow  $X_{t+1}$  of  $t + 1$ .  
Process:  
(1) for  $i = 1, 2, \dots, n$  do  
(2)  $\{Z_{(D,t-1)}, Z_{(D,t)}\} \leftarrow \text{Multi\_GCN}(\{X_1^{V_i}, X_2^{V_i}, \dots, X_t^{V_i}\}, A_D)$  by equation (7)  
(3)  $\{Z_{(C,t-1)}, Z_{(C,t)}\} \leftarrow \text{Multi\_GCN}(\{X_1^{V_i}, X_2^{V_i}, \dots, X_t^{V_i}\}, A_C)$  by equation (8)  
(4)  $\{\Delta Z_D, \Delta Z_C\} \leftarrow \text{Diff\_GCN}(\{(Z_{(D,t-1)}, Z_{(D,t)}), (Z_{(C,t-1)}, Z_{(C,t)})\})$  by equations (10) and (11)  
(5)  $Z \leftarrow \text{Attention}(Z_{(D,t)}, Z_{(C,t)}, \Delta Z_D, \Delta Z_C)$   
(6)  $X_{t+1}^{V_i} \leftarrow \text{FC}(\text{LSTM}(Z))$   
(7) end  
(8)  $\{X_{t+1}|X_{t+1}^{V_1}, X_{t+1}^{V_2}, \dots, X_{t+1}^{V_n}\}$  (Forecasted passenger flow of all stations)

ALGORITHM 1: MDGCN.

## 4. Results and Discussion

**4.1. Experimental Settings.** This model is implemented in Tensorflow and Keras. All experiments were run on a graphics processing unit (GPU) platform with an NVIDIA GeForce GTX 1070 Ti graphics card and 8GB GPU memory. We train our model using Adam optimizer with a learning rate of 0.0001. The training epoch is 150. Two kinds of evaluation metrics are adopted, including root mean squared errors (RMSE), mean absolute errors (MAE), and the evaluation indicators which are shown in

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}, \quad (17)$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |(y_i - \hat{y}_i)|, \quad (18)$$

where  $y_i$  is the true value,  $\hat{y}_i$  is the predicted value, and  $n$  is the number of samples.

**4.2. Dataset.** We used two real-world datasets from China to evaluate our model, i.e., data collected from Xiamen and Shanghai subway Automatic Fare Collection (AFC) System.

SH dataset was collected from 1 April to 31 April 2015. This dataset includes records from 14 lines and 313 stations. The example of this dataset is shown in Table 1. Each record contains the card number (ID), travel date (DATE), card-swiping time (TIME), the name of stations (STATION\_NAME), the number of lines (LINE\_ID), and TPYE. Passenger flow is calculated by formula (19).  $x_{t-b}^a$  represents the number of passenger flows of station  $a$  in period  $t - b$ . Here,  $a$  is determined by "STATION\_NAME" and  $t - b$  is determined by "DATE" and "TIME."

XM dataset was collected from 1 December to 31 December 2019. This dataset includes records from two lines and 52 stations. The example of this dataset is shown in Table 2. We use the same procedure of data preprocessing to gain passenger flow.

$$X = \begin{pmatrix} x_t^1 & x_{t-1}^1 & x_{t-b}^1 & \dots & x_{t-m-1}^1 \\ x_t^2 & x_{t-1}^2 & x_{t-b}^2 & \dots & x_{t-m-1}^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_t^n & x_{t-1}^n & x_{t-b}^n & \dots & x_{t-m-1}^n \end{pmatrix}. \quad (19)$$

TABLE 1: The sample of Shanghai dataset.

ID	DATE	TIME	STATION_NAME	LINE_ID	FARE	TYPE
6021**128	2015/4/17	9:34:30	Xinzhuang	1	2	Discount
2801**273	2015/4/17	18:23:59	Dalian Road	4	1	Nondiscount
1014**819	2015/4/17	9:16:18	Guanglan Road	2	4	Discount

TABLE 2: The sample of Xiamen dataset.

ID	DATE	TIME	STATION_ID	STATION_NAME	LINE_ID	KHTYPE
17**25	20191224	51090	21	DongFu	2	E-card
82**47	20191210	73725	11	HuoJuYuan	1	E-card
94**60	20191231	68397	26	GuanYin	2	E-card

4.3. *Experiment Results.* In this section, we conducted four main parts of experiments to demonstrate the validity of the proposed model and the reasonableness of each module.

4.3.1. *Evaluation of the Number of GCN Layers.* To determine the optimal value of the number of GCN stacks, we conduct experiments at different stacking values to obtain the corresponding prediction values, and the prediction results are shown in Figure 6. We found that the experiment worked best when  $N = 3$ , so we set the number of stacks to 3 in the experiment.

4.3.2. *The Results in Baselines.* To evaluate the competitive performance of the proposed method (i.e., MDGCN), we compared it with two types of models, covering the most basic mathematical-statistical models and the deep learning models. All baselines are optimized to output the best performance.

- (i) HA [27]: it means historical average model, which uses the average of several previous periods as the prediction. In this paper, we use the average of the last ten periods as the forecast value.
- (ii) ARIMA [28]: we used SPSS software to make predictions so that we could get the best predictions.
- (iii) SVR: we used support vector regression (SVR) to predict passenger flow. Here, the kernel function is rbf, epsilon is 0.2.
- (iv) GBDT: we used gradient boosting decision tree (GBDT) to predict passenger flow. Here,  $n\_estimators$  is 100,  $min\_samples\_split$  is 2, and  $learning\_rate$  is 0.1.
- (v) LSTM: Ma et al. [8] used this method to predict traffic flow. In this paper, we use the same method to predict passenger flow. We used a stacked LSTM which has 128 and 276 neurons for the first and second layers.
- (vi) GCN + LSTM: Chai et al. [17] used this method to predict bike flow. Likewise, we use the same method to predict passenger flow. We used a 2-layer GCN and a layer of LSTM which has 64 neurons.

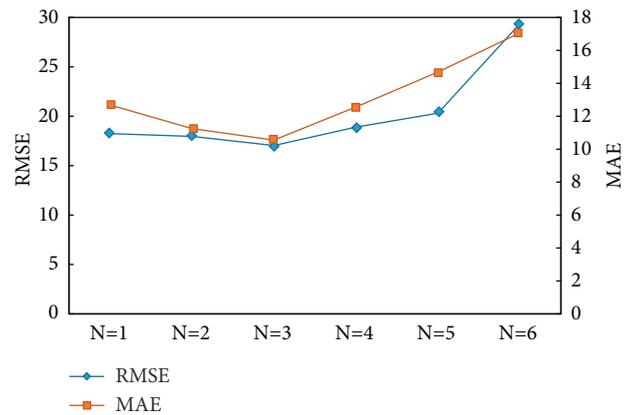


FIGURE 6: Performance with different number of GCN layer.

- (vii) ResLSTM: it refers to the method in [29], which combines ResNet and GCN with LSTM to predict the traffic flow. Here we use the same setting as it is without branch 4.
- (viii) Conv-GCN: it refers to the method in [1], which combines the 3D-Conv with GCN to predict the traffic flow. Also, we use the same setting as is.

The results are shown in Table 3. The following trends can be seen from the results. Firstly, deep learning models (including LSTM, Conv-GCN, ResLSTM, GCN + LSTM, and MDGCN) have a better performance than traditional mathematical-statistical (including HA and ARIMA) and shallow machine learning models (including SVR and GBDT). This result shows that deep learning methods can better capture nonlinear spatiotemporal correlations. Second, the variant of recurrent neural network (including LSTM and ResLSTM) outperforms machine learning methods (including SVR and GBDT). It indicates these time-series models can capture temporal correlation deeply. By comparison, the spatial deep learning-based models (including Conv-GCN, GCN + LSTM, and MDGCN) outperform the variant of LSTM. It demonstrates the effectiveness of spatial correlation in passenger flow prediction. Also, MDGCN mines the relationship of all the stations deeply to have the best performance compared to all baselines. Thirdly, almost all the models behave much better on XM than SH. The biggest reason must be the complexity of the dataset. SH dataset has more lines and stations than

TABLE 3: The performance in different baselines.

Model	SH					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
HA	69.812	89.210	95.574	35.954	49.034	54.601
ARIMA	51.044	81.012	156.256	29.057	44.257	82.964
SVR	30.454	36.214	39.754	17.207	19.903	21.169
GBDT	29.935	39.364	32.542	16.561	20.001	17.342
LSTM	28.059	26.541	29.354	15.705	14.672	15.584
Conv-GCN	22.154	21.097	25.645	11.254	14.360	14.391
ResLSTM	26.387	29.548	29.642	14.087	15.265	16.005
GCN + LSTM	20.165	21.364	24.608	13.904	14.367	13.927
MDGCN	17.041	20.962	22.354	10.552	12.786	13.452

Model	XM					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
HA	67.241	85.455	95.147	35.264	47.751	53.813
ARIMA	50.695	79.568	175.265	28.564	43.561	95.556
SVR	29.365	35.524	36.524	16.454	18.428	19.658
GBDT	33.635	37.652	50.562	17.058	19.952	26.895
LSTM	28.545	26.421	30.241	14.214	16.754	19.241
Conv-GCN	19.421	20.145	24.245	12.256	13.978	15.842
ResLSTM	24.007	23.557	26.352	14.545	15.812	18.731
GCN + LSTM	18.954	19.145	22.296	11.925	10.770	13.754
MDGCN	16.713	16.927	18.560	9.924	10.247	11.155

TABLE 4: The evaluation results of spatial relationship reconstruction.

Model	SH					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
MDGCN-A	35.548	30.674	29.541	19.066	16.078	15.597
MDGCN-AD	25.685	35.547	36.545	14.950	16.635	18.654
MDGCN-AC	20.365	26.654	23.315	11.027	14.028	12.389
MDGCN	17.041	20.962	22.354	10.552	12.786	13.452

Model	XM					
	RMSE			MAE		
	10 min	15 min	30 min	10 min	15 min	30 min
MDGCN-A	27.241	27.624	30.148	18.279	18.698	23.039
MDGCN-AD	22.148	55.512	27.539	14.208	26.384	14.257
MDGCN-AC	18.214	19.012	22.351	10.852	11.109	13.691
MDGCN	16.713	16.927	18.560	9.924	10.247	11.155

XM. Thus, the traffic condition patterns in Shanghai would be much more complex than in Xiamen. Finally, MDGCN has the best performance in these two datasets. Compared with the optimal baseline, it reduced error by 1% to 15%. This shows that it can get better performance by capturing various correlations (including spatial and temporal) and deeply mining the historical passenger flow.

*4.3.3. Evaluation of Spatial Relationship Reconstruction of MDGCN.* In the third experiment, we try to evaluate the effectiveness of spatial relationship reconstruction. Specifically, we compare the performance of the following three variants of MDGCN. Here, AM represents the adjacent matrix that is put into the Multi-GCN layer.

- (i) MDGCN-A: it concludes only the most basic adjacency matrix (i.e.,  $AM = A$ )

- (ii) MDGCN- $A_D$ : it concludes only the physical topology matrix based on the adjacency matrix with distance inverse recoding (i.e.,  $AM = A_D$ )
- (iii) MDGCN- $A_C$ : it concludes only the topological matrix based on the passenger space (i.e.,  $AM = A_C$ )
- (iv) MDGCN: it combines the topological matrix of both physical space and passenger space (i.e.,  $AM = A_D \cup A_C$ )

The experiment results are shown in Table 4. On the one hand, MDGCN has the best overall performance and MDGCN-A has the worst. It demonstrates the importance and effectiveness of recoding. On the other hand, MDGCN- $A_C$  outperforms MDGCN-A and MDGCN- $A_D$ , and it has almost the same level of accuracy as MDGCN. It suggests that the topology of the passenger space more accurately describes the correlations between stations.

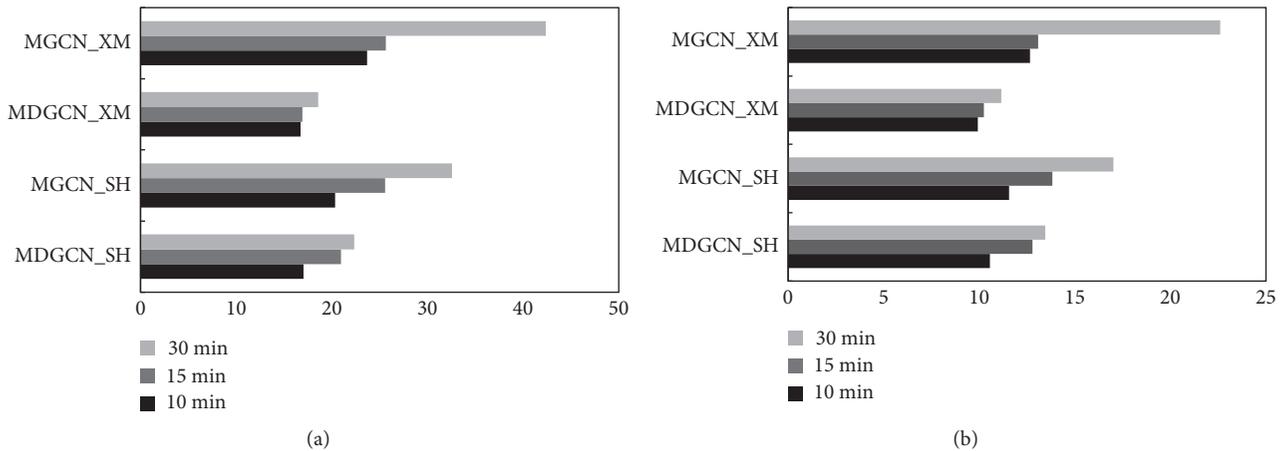


FIGURE 7: The evaluation of Diff-GCN under different time intervals. (a) RMSE. (b) MAE.

**4.3.4. Evaluation of the Diff-GCN Layer of MDGCN.** In the last experiment, we evaluate the performance of the layer of Diff-GCN. We compare the performance of MDGCN with its variant called MGCN which does not conclude the module of Diff-GCN. The experiment results are shown in Figure 7.

It is clear from the results on these two datasets are consistent with each other. The layer of Diff-GCN reduces the prediction error because it can fully extract the trend information among passenger flow.

## 5. Conclusions

In this paper, we propose a deep learning model called MDGCN to predict passenger flow. We construct the Multi-GCN layer to extract heterogeneous correlations under two spatial relationships, and we use the Diff-GCN layer to extract the changing trend of heterogeneous features to fully capture the spatiotemporal correlations. Eventually, it is validated in the actual datasets and obtained better results. However, there are still many points for improvement in the above work, for example, how to construct dynamic connectivity of stations using historical passenger flow data and how to consider the impact of various factors such as weather and unexpected accidents on the prediction results. Therefore, future work will focus on addressing these issues.

## Data Availability

The original data set cannot be provided due to the confidentiality of the data.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the General Project of Natural Science Foundation of Fujian Province of China (2017J01111), the Science and Technology Plan Guiding

Project of Fujian Province (2020H0016), the National Nature Science Foundation of China (61802133), and the Fundamental Research Funds for the Central Universities (ZQN-910).

## References

- [1] J. Zhang, F. Chen, Y. Guo, and X. Li, "Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit," *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1210–1217, 2020.
- [2] L. M. Moreira, J. Gama, M. Ferreira, J. M. Mendes, and L. Damas, "Predicting taxi-passenger demand using streaming data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 3, pp. 1393–1402, 2013.
- [3] H. Su and S. Yu, "Hybrid GA based online support vector machine model for short-term traffic flow forecasting," in *Proceedings of the International Workshop on Advanced Parallel Processing Technologies*, pp. 743–752, Guangzhou, China, November 2007.
- [4] J. Luo, J. Li, J. Wang, Z. Jiang, and Y. Chen, "Deep attributes from context-aware regional neural codes," 2015, <https://arxiv.org/abs/1509.02470v1>.
- [5] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Hilton San Francisco Union Square, CA, USA, February 2017.
- [6] K. Zhang, Z. Liu, and L. Zheng, "Short-term prediction of passenger demand in multi-zone level: temporal convolutional neural network with multi-task learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1480–1490, 2020.
- [7] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *Proceedings of the 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, Wuhan, China, November 2016.
- [8] 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.
- [9] D. Yang, K. Chen, M. Yang, and X. Zhao, "Urban rail transit passenger flow forecast based on LSTM with enhanced long-

- term features,” *IET Intelligent Transport Systems*, vol. 13, no. 10, pp. 1475–1482, 2019.
- [10] R. L. Abduljabbar, H. Dia, and P. W. Tsai, “Unidirectional and bidirectional LSTM models for short-term traffic prediction,” *Journal of Advanced Transportation*, vol. 2021, Article ID 5589075, 16 pages, 2021.
- [11] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, “Traffic graph convolutional recurrent neural network: a deep learning framework for network-scale traffic learning and forecasting,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 11, pp. 4883–4894, 2019.
- [12] L. Lin, Z. He, and S. Peeta, “Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach,” *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.
- [13] T. Bogaerts, A. D. Masegosa, J. S. Z. Angarita, E. Onieva, and P. Hellinckx, “A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data,” *Transportation Research Part C: Emerging Technologies*, vol. 112, pp. 62–77, 2020.
- [14] L. Zhao, Y. Song, and C. Zhang, “T-GCN: a temporal graph convolutional network for traffic prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.
- [15] M. Lv, Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji, “Temporal multi-graph convolutional network for traffic flow prediction,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 6, pp. 3337–3348, 2021.
- [16] X. Geng, Y. Li, L. Wang et al., “Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 1, pp. 3656–3663, Honolulu, HI, USA, February 2019.
- [17] D. Chai, L. Wang, and Q. Yang, “Bike flow prediction with multi-graph convolutional networks,” in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 397–400, Seattle, WA, USA, November 2018.
- [18] Y. He, Y. Zhao, H. Wang, and K. L. Tsui, “G. C.-L. S. T. M.: A deep spatiotemporal model for passenger flow forecasting of high-speed rail network,” in *Proceedings of the 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1–6, Rhodes, Greece, September 2020.
- [19] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, “Multistep speed prediction on traffic networks: a deep learning approach considering spatio-temporal dependencies,” *Transportation Research Part C: Emerging Technologies*, vol. 105, pp. 297–322, 2019.
- [20] Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, “Graph wavenet for deep spatial-temporal graph modelling,” 2019, <https://arxiv.org/abs/1906.00121>.
- [21] C. Park, C. Lee, and H. Bahng, “S. T. G. R. A. T.: A spatio-temporal graph attention network for traffic forecasting,” 2019, <https://arxiv.org/abs/1911.13181>.
- [22] A. Vaswani, N. Shazeer, and N. Parmar, “Attention is all you need,” in *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, pp. 5998–6008, CA, USA, December 2018.
- [23] S. Hao, D.-H. Lee, and D. Zhao, “Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system,” *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 287–300, 2019.
- [24] Z. Guo, X. Zhao, and Y. Chen, “Short-term passenger flow forecast of urban rail transit based on GPR and KRR,” *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1374–1382, 2019.
- [25] X. Wang, M. Zhu, D. Bo, P. Cui, C. Shi, and J. Pei, “Am-Gcn: Adaptive multi-channel graph convolutional networks,” in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1243–1253, CA, USA, August 2020.
- [26] Y. Wang, H. Yin, H. Chen, T. Wo, J. Xu, and K. Zheng, “Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modelling,” in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1227–1235, Anchorage, AK, USA, August 2019.
- [27] J. Liu and W. Guan, “A summary of traffic flow forecasting methods,” *Journal of Highway and Transportation Research and Development*, vol. 3, pp. 82–85, 2004.
- [28] M. M. Hamed, H. R. A. Maseid, and Z. M. B. Said, “Short-term prediction of traffic volume in urban arterials,” *Journal of Transportation Engineering*, vol. 121, no. 3, pp. 249–254, 1995.
- [29] J. Zhang, F. Chen, Z. Cui, Y. Guo, and Y. Zhu, “Deep learning architecture for short-term passenger flow forecasting in urban rail transit,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7004–7014, 2021.