

# A Metro Passenger Flow Forecasting Model Based On Time-series Evolving Interaction Graph Network

Jianming Han<sup>1, a</sup>, Pei Li<sup>2, b\*</sup>, Long Li<sup>1, c</sup>, Yingdi Li<sup>2, d</sup>, and Hantao Zhao<sup>3, e</sup>

<sup>1</sup> Operation Management Center of Suzhou Rail Transit Group Co., Ltd, Suzhou 215000, Jiangsu, China; <sup>2</sup> School of Transportation, Southeast University, Nanjing 211102, Jiangsu, China; <sup>3</sup> School of Cyber Science and Engineering, Southeast University, Nanjing 211102, Jiangsu, China.

<sup>a</sup> hanjianming@sz-mtr.com, <sup>b</sup> lipei\_seu@163.com, <sup>c</sup> lilong@sz-mtr.com, <sup>d</sup> yingdi.li@qq.com, <sup>e</sup> htzhao@seu.edu.cn

**Abstract.** Passenger flow forecasting is an important task in metro operation management. In order to achieve more accurate metro passenger flow forecasting, this paper proposes a metro passenger flow forecasting model based on time-series evolution interaction graph. First, by introducing two kinds of inter-station interaction graphs, namely connectivity graph and temporal correlation graph, to capture the potential interaction relationship among metro passenger flow stations. Then, by using the time-series evolving graph, the weights of the graph convolutional neural network are dynamically evolved in time series. Finally, taking Suzhou Metro as an example, the short-term passenger flow of the metro is forecasted. The experimental results show that the Root Mean Squared Error (RMSE) of this model is 34.17, the Mean Absolute Error (MAE) is 16.35, the R-Squared is 0.94, the Mean Absolute Percent Error (MAPE) is 0.21. All evaluation metrics are better than the baseline models, thus verifying the effectiveness and applicability of the metro passenger flow forecasting model based on the time series evolution interaction graph.

**Keywords:** Metro passenger flow forecasting; graph convolutional network; time-series evolving graph; interaction graph.

## 1. Introduction

As a primary pillar of urban public transportation, the metro plays a crucial role in the daily travel activities of city residents. However, with the continuous development of urbanization and the ever-growing urban population, imbalances in passenger distribution along metro routes have emerged, leading to a series of operational issues and safety concerns<sup>[1,2]</sup>. The timely prediction results empower metro operating management to implement proactive control strategies, disseminate information, and enable passengers to anticipate metro flow and congestion conditions in advance<sup>[3-5]</sup>. Therefore, accurate short-term metro passenger flow prediction is crucial for both metro operating departments and passengers.

This paper proposes an interaction graph to better capture the latent interaction relationships in metro passenger flow. Additionally, a temporal evolution graph is introduced to dynamically update the weight matrices of Graph Convolutional Neural Networks based on real-time dynamic passenger flow information. This establishes a metro passenger flow prediction model based on the temporal evolution interaction graph. Experimental validation using the Suzhou metro smart card dataset demonstrates the effectiveness of the proposed model.

## 2. Model Construction

### 2.1 Long Short-Term Memory Neural Network

The Long Short-Term Memory (LSTM) is a special type of recurrent neural network capable of capturing long-term dependency relationships while avoiding the issues of gradient vanishing or exploding. The fundamental unit of LSTM is the memory cell, composed of an input gate, forget gate, output gate, and the memory cell itself.

#### 1) Forget Gate

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \#(1)$$

where  $h_{t-1}$  represents the output information from the previous time step,  $x_t$  is the input information at the current time step,  $W_f$  is the weight matrix,  $b_f$  is the bias vector, and  $\sigma$  denotes the sigmoid activation function. In the forget gate, the sigmoid activation function is selected, with the output ranging between 0 and 1:

$$\sigma = \frac{1}{1 + e^{-x}} \#(2)$$

### 2) Input Gate

The input gate determines the amount of new information to be input. First, the information  $\tilde{C}_t$  to be input is calculated:

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C), \#(3)$$

where  $W_C$  is the weight matrix,  $b_C$  is the bias vector, and  $\tanh$  represents the nonlinear activation function, producing output values in the range  $[-1, 1]$ :

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \#(4)$$

Simultaneously,  $i_t$  is calculated to determine which information to input:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \#(5)$$

where  $W_i$  is the weight matrix,  $b_i$  is the bias vector, and  $\sigma$  is the sigmoid activation function. The forget vector  $f_t$  and the input vector  $i_t$  obtained from the forget gate and input gate, respectively, along with the previous input information  $C_{t-1}$  and the current input information  $\tilde{C}_t$ , combine to obtain the final updated information  $C_t$ :

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \#(6)$$

### 3) Output Gate

After processing through the forget and input gates, the updated information needs to be filtered and output. Firstly,  $o_t$  is calculated to determine which information to output:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \#(7)$$

where  $W_o$  is the weight matrix,  $b_o$  is the bias vector.

## 2.2 Graph Convolutional Neural Network

Graph Convolutional Neural Network (GCN) is a type of deep learning model based on graphs, primarily used for handling data with complex structures. GCN can extract feature representations of nodes in a graph, considering the interrelations between nodes. The fundamental building block of GCN is the graph convolutional layer, taking as input the node features on the graph and the edge features between nodes, and producing new feature representations for each node on the graph. A two-layer Graph Convolutional Neural Network structure is illustrated in Figure 1.

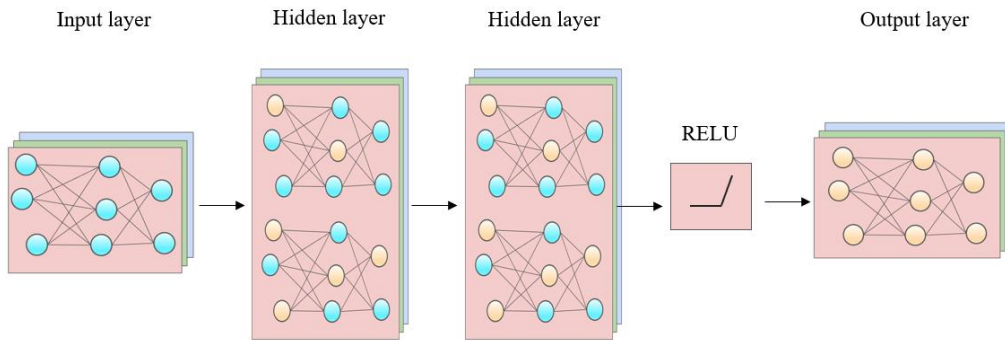


Fig. 1 Graph convolutional neural network architecture

Consider a graph  $G = \{V, E\}$ , consisting of  $N$  nodes, where node  $v_i \in V$ , edge  $e_i \in E$ , the adjacency matrix  $A \in R^{N \times N}$  and feature matrix  $H \in R^{N \times C}$ . To retain a node's own information, we replace  $A$  with  $\tilde{A}$ , where  $\tilde{A} = A + I$  and  $I$  is the identity matrix. Additionally, we calculate the

degree matrix  $I$  for  $\tilde{A}$ . The propagation formula between layers of the graph convolutional neural network is:

$$H^{(l+1)} = \sigma\left(\tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}H^{(l)}W^{(l)}\right), \#(8)$$

where  $H^{(l)}$  is the feature matrix for the  $l$ -th layer,  $W^{(l)}$  is the weight matrix for the  $l$ -th layer, and  $\sigma$  is the nonlinear activation function.

In summary, for a Graph Convolutional Neural Network with two layers, the formula is:

$$Z = f(H, A) = \text{softmax}(\hat{A} \text{RELU}(\hat{A}HW^{(0)})W^{(1)}), \#(9)$$

where  $\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}$ , and softmax and ReLU are nonlinear activation functions defined as:

$$\text{ReLU}(x) = \begin{cases} \max(0, x) & , x \geq 0 \\ 0 & , x < 0 \end{cases} \#(10)$$

$$\text{softmax}(x) = \frac{e^x}{\sum_{x=1}^N e^x} \#(11)$$

## 2.3 Temporal Evolutionary Interaction Graph

### 1) Temporal Evolutionary Graph

This paper proposes a temporal evolutionary graph to evolve the weight parameters of each GCN model at each time step based on the input temporal metro flow information. LSTM is used for parameter concatenation, with the model's hidden state using the parameters from the previous time step, and GRU taking the current time step's node representation as input. The formulas are as follows:

$$W_t^{(l)} = \text{LSTM}(H_t^{(l)}, W_{t-1}^{(l)}), \#(12)$$

$$H_t^{(l+1)} = \text{GCN}(A_t, H_t^{(l)}, W_t^{(l)}), \#(13)$$

where  $A_t$  is the adjacency matrix,  $W_t^{(l)}$  is the weight matrix of the  $l$ -th layer at time step  $t$ ,  $H_t^{(l)}$  is the hidden state of the  $l$ -th layer at time step  $t$ , where  $H_t^{(0)}$  is the input metro flow matrix at time  $t$ .

### 2) Interaction Graph

**Connectivity Graph:** This graph represents the impact of walking duration between stations on passenger entrance choices. The reciprocal of walking duration is used to indicate the strength of connectivity between stations. To ensure that the values of the connectivity graph adjacency matrix  $A_c$  fall within the range of 0 to 1, introduce the parameter  $\gamma$  for normalization and set  $\gamma = \min_{a \neq b} t(a, b)$ . Therefore,  $A_c(a, b)$  is calculated as follows:

$$A_c(a, b) = \frac{\gamma}{t(a, b)} \#(14)$$

**Temporal Correlation Graph:** Conducting statistical analysis on passenger flows along the time series and computing correlation coefficients proves effective in capturing the latent time-related correlations within the passenger flows. Let  $Q_a = \{f_{a,1}, f_{a,2}, \dots, f_{a,t}, \dots, f_{a,N}\}$  represent the passenger flow time series at station  $a$ , where  $f_{a,t}$  is the passenger flow at station  $a$  at time  $t$ ,  $N$  is the length of the time series, and  $M_{a,t}$  is the average passenger flow at station  $a$ .  $A_t(a, b)$  represents the time correlation between stations  $a$  and  $b$ , calculated as follows:

$$A_t(a, b) = \frac{\sum_{t=1}^T (f_{a,t} - M_{a,t})(f_{b,t} - M_{b,t})}{\sqrt{\sum_{t=1}^T (f_{a,t} - M_{a,t})^2} \sqrt{\sum_{t=1}^T (f_{b,t} - M_{b,t})^2}} \#(15)$$

### 3) Temporal Evolutionary Interaction Graph

Building upon the aforementioned temporal evolution graph and the introduction of the interaction graph, this paper presents a temporal evolution interaction graph for metro passenger flow prediction, as show in Figure 2. The adjacency matrix of the graph convolutional neural network is formed by the combination of the connectivity graph, temporal correlation graph, and physical adjacency matrix. This facilitates the comprehensive extraction of topological features and hidden interaction relationships within metro passenger flows. The GCN is configured as a dual-layer graph convolution, with the weights at each time step evolving through LSTM-based updates from the preceding time step. The selected adjacency matrices include the physical adjacency matrix  $A_p$  (where elements are 1 if two stations are adjacent and 0 otherwise), the connectivity graph adjacency matrix  $A_c$ , and the temporal correlation adjacency matrix  $A_t$ .

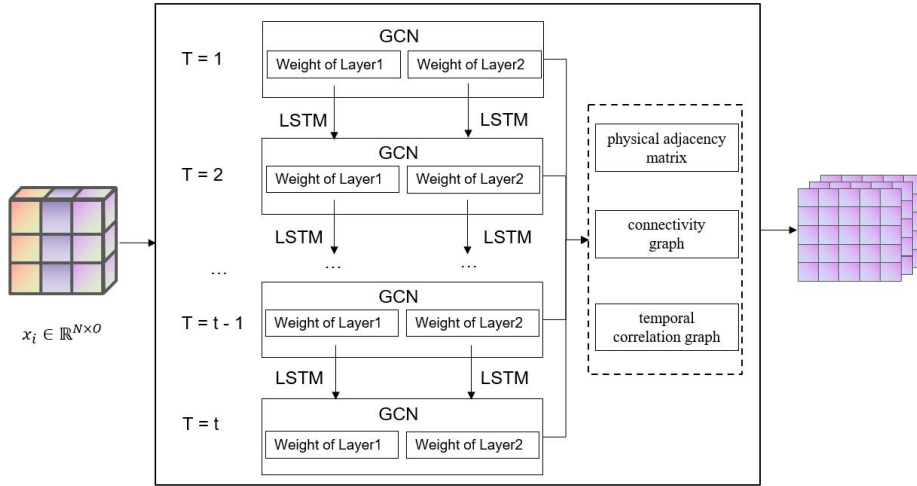


Fig. 2 Model structure of temporal evolution interaction graph

### 3. Experiment Design

#### 3.1 Dataset Description

This study focuses on short-term prediction of metro entrance passenger flow, using Suzhou Metro as the research subject. The dataset employed consists of AFC card-swiping data from Suzhou Metro for the entire month of August 2018, spanning 31 days. As of August 2018, Suzhou Metro operates three metro lines: Line 1, Line 2, and Line 4 (including a branch), comprising a total of 93 metro stations.

The dataset is processed using 15-minute time intervals. The input for the model consists of the entrance passenger flow for all stations over the past 8 time steps, while the output comprises the entrance passenger flow for all stations in the next time step. Finally, the dataset is divided into training, validation, and testing sets in a 6:2:2 ratio.

#### 3.2 Experimental Results Analysis

In this section, three classical metro passenger flow prediction models, Historical Average (HA), TGCN<sup>[6]</sup>, and ASTGCN<sup>[7]</sup>, are selected as baseline models for comparison with the proposed model to demonstrate its effectiveness.

Table 1. Comparison of forecasting results between the proposed model and the baseline models

Model	RMSE	MAE	R2	MAPE
HA	72.87	36.79	0.71	0.38
TGCN	48.18	23.45	0.87	0.29
ASTGCN	41.25	20.74	0.90	0.24
Proposed model	34.17	16.35	0.94	0.21

In the experimental testing set, this paper compares the proposed model with four baseline models. To comprehensively evaluate passenger flow predictions, four metrics—RMSE, MAE, R2, and MAPE—are calculated and presented in Table 1. The baseline HA model exhibits the poorest prediction performance with a MAPE of 0.38. T-GCN and PVGCN outperform HA, but the model proposed in this paper, based on the time-evolving interaction graph, achieves the best performance across all three evaluation metrics: RMSE of 34.17, MAE of 16.35, R2 of 0.94, and MAPE of 0.21. These experimental results demonstrate that the proposed model can effectively capture potential interaction relationships among metro passenger flows, showcasing excellent passenger flow prediction capabilities.

## 4. Conclusion

This paper proposes a passenger flow prediction model based on the time-evolving interaction graph for short-term metro entrance passenger flow prediction. To address the insufficient feature extraction in existing research, two graphs are proposed to capture potential interactions between stations: the connectivity graph and the time-related graph. When designing interaction graphs, this paper uses travel information provided by network maps, considering the impact of walking time on passenger station choices and the potential correlation in passenger flow time series between different stations. Additionally, a new deep learning-based model is designed to apply interaction graphs and time-evolving graphs in metro passenger flow prediction. This model utilizes time-evolving graphs to update graph convolutional neural network weights, allowing model parameters to be updated with metro passenger flow data input. Finally, using Suzhou metro data as an example, the model's performance is evaluated. Experimental results indicate that compared to baseline models, the proposed model performs better and achieves more accurate metro passenger flow predictions.

## Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities under Grant 2242021R10093.

## References

- [1] MENG Pinchao, LI Xueyuan, JIA Hongfei, et al. Short-time rail transit passenger flow real-time prediction based on moving average[J]. Journal of Jilin University Engineering and Technology Edition, 2018, 48(2): 448-453
- [2] Nigam A, Srivastava S. Hybrid deep learning models for traffic stream variables prediction during rainfall[J]. Multimodal Transportation, 2023, 2(1): 100052.
- [3] ZHU Caihua, SUN Xiaoli, LI Peikun, et al. Short-term urban rail transit passenger flow prediction based on incorporating station classification and data noise reduction[J]. Journal of Railway Science and Engineering, 2022, 19(08): 2182-2192.
- [4] Ma Z, Zhang P. Individual mobility prediction review: Data, problem, method and application[J]. Multimodal Transportation, 2022, 1(1): 100002.
- [5] Yan R, Wang S. Integrating prediction with optimization: Models and applications in transportation management[J]. Multimodal Transportation, 2022: 100018.
- [6] Zhao L , Song Y , Zhang C ,et al. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction[J]. 2018.
- [7] Guo S , Lin Y , Feng N ,et al. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting[C]. National Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence (AAAI), 2019.