International Journal of
Geo-Information

Article

A Hybrid GLM Model for Predicting Citywide Spatio-Temporal Metro Passenger Flow

Yong Han [1,2,*], Tongxin Peng [1,2], Cheng Wang [3,4], Zhihao Zhang [1,2] and Ge Chen [1,2]

[1] College of Information Science and Engineering, Ocean University of China, No. 238, Songling Road, Qingdao 266100, China; tongxinp@stu.ouc.edu.cn (T.P.); zhangzhihao3974@stu.ouc.edu.cn (Z.Z.); gechen@ouc.edu.cn (G.C.)

[2] Laboratory for Regional Oceanography and Numerical Modeling, Qingdao National Laboratory for Marine Science and Technology, No. 1, Wenhai Road, Qingdao 266237, China

[3] Center of Grassroots Governance Led by the Chinese Communist Party in Shibei District, No. 161, Tailiu Road, Qingdao 266000, China; wc8387@stu.ouc.edu.cn

[4] Big Data Development Bureau of Shibei District, No.161, Tailiu Road, Qingdao 266000, China

* Correspondence: yonghan@ouc.edu.cn

Abstract: Accurate prediction of citywide short-term metro passenger flow is essential to urban management and transport scheduling. Recently, an increasing number of researchers have applied deep learning models to passenger flow prediction. Nevertheless, the task is still challenging due to the complex spatial dependency on the metro network and the time-varying traffic patterns. Therefore, we propose a novel deep learning architecture combining graph attention networks (GAT) with long short-term memory (LSTM) networks, which is called the hybrid GLM (hybrid **G**AT and **L**STM **M**odel). The proposed model captures the spatial dependency via the graph attention layers and learns the temporal dependency via the LSTM layers. Moreover, some external factors are embedded. We tested the hybrid GLM by predicting the metro passenger flow in Shanghai, China. The results are compared with the forecasts from some typical data-driven models. The hybrid GLM gets the smallest root-mean-square error (RMSE) and mean absolute percentage error (MAPE) in different time intervals (TIs), which exhibits the superiority of the proposed model. In particular, in the TI 10 min, the hybrid GLM brings about 6–30% extra improvements in terms of RMSE. We additionally explore the sensitivity of the model to its parameters, which will aid the application of this model.

Keywords: deep learning; short-term passenger flow prediction; graph attention network; long short-term memory network; hybrid model

check for updates

Citation: Han, Y.; Peng, T.; Wang, C.; Zhang, Z.; Chen, G. A Hybrid GLM Model for Predicting Citywide Spatio-Temporal Metro Passenger Flow. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 222. https://doi.org/10.3390/ijgi10040222

Academic Editors: José F. Vicent and Wolfgang Kainz

Received: 27 January 2021
Accepted: 1 April 2021
Published: 3 April 2021

1. Introduction

People constantly interact with the urban space through various spatio-temporal activities, such as taking the subway, driving, and walking [1]. In the big data era, the rapid proliferation of mobile sensors and Internet technologies continuously generates an exceptionally large amount of spatio-temporal data, which offers unprecedented opportunities for constructing intelligent transportation systems (ITS). In particular, short-term metro passenger flow prediction is an important part of ITS. Accurate prediction of passenger flow can help urban managers to fine-tune travel behaviors, reduce passenger congestion, and enhance the service quality of metro system [2]. From a broader point of view, metro passenger flow prediction helps to optimize traffic efficiency via alleviating the imbalance of transport capacity across the city. Therefore, developing an effective framework for predicting passenger flow in a citywide metro network is essential.

Due to its great practical value, passenger flow prediction has been extensively investigated. Existing solutions can be classified into three categories: statistical methods, machine learning (ML) methods [3], and deep learning (DL) methods. Statistical methods are simple but cannot capture non-linear features. ML methods improve the drawbacks of statistical methods but are still incapable of processing raw spatio-temporal data. When constructing an ML-based model, feature extractors require precise engineering and substantial domain knowledge to transform raw data into a proper internal representation. This procedure is called feature engineering. With regard to big data, the feature engineering procedure is particularly challenging. Compared with ML methods, DL methods can automatically build up feature engineering and accept raw input to make an end-to-end prediction, which can learn more complex non-linear characters and gain better generalization ability. DL methods are the most widely used solution for passenger flow prediction. As metro passenger flow prediction is a time series processing problem, recurrent neural network (RNN) is effective for the task [4,5]. In addition, it has become a popular trend to exploit RNNs in combination with the convolutional neural networks (CNNs) for traffic flow prediction [6,7], owing to the ability of CNNs to mine spatial dependency. However, CNNs are designed for spatial structure in Euclidean space (e.g., 2D images and regular grids), so it cannot fully adapt the complex topological structure of a metro network. Aiming at this problem, graph modeling on spatio-temporal data has been in the spotlight. Several works have studied Graph Neural Networks (GNNs) for capturing topological spatial correlation [8–10]. However, existing graph-structure-based approaches still have the following gaps:

1. Most graph-structure-based approaches are based on the Graph Conventional Network (GCN) [11], which is operated in the spectral domain. Due to the use of the Laplacian matrix, GCN requires the network to be symmetric. However, there always exists some asymmetric networks in a city. The network in a city whose graph structure is asymmetric can be defined as an asymmetric network. For example, a road network with one-way and two-way streets can be described as an asymmetric network. A GCN-based structure cannot be used in this case.
2. Most graph-structure-based methods ignore the improvement of the adjacent matrix. In other words, they only care about the effect of the adjacent nodes but ignore the nodes located a little further.
3. Some graph-structure-based models only capture the spatial dependency but ignore the temporal dependency and external factors.

To overcome the abovementioned issues, we propose a hybrid DL model for short-term metro passenger flow prediction by integrating graph attention networks (GATs) and long short-term memory (LSTM) networks. The proposed model is called the hybrid GLM (hybrid **G**AT and **L**STM **M**odel). Inspired by Petar Veličković [12], we introduce the GAT to solve the problem that GCN cannot be applied to asymmetric networks in a city. GAT captures the spatial dependency of adjacent nodes by calculating the graph attention coefficient between nodes, which is beneficial to asymmetric graphs. In addition to GAT, we combine LSTM networks for modeling the temporal dependency of metro passenger flow. External factors related to metro passenger flow include weather conditions, air quality, weekends, holidays and events, etc. We add these factors into another LSTM layer to improve the accuracy of the entire model. The main contributions of this paper are as follows:

1. We propose a hybrid graph-structure-based model to predict the short-term metro passenger flow. The GAT structure in the proposed model can capture the complex topological dependency. Besides, GAT puts more focus on nodes, which means it can solve the problem that GCN cannot be used in asymmetric networks. Moreover, we improve the adjacent matrix in the GAT structure for modeling the nodes located a little further.
2. We construct a novel framework to jointly model the spatial, dynamic temporal and external dependencies in metro flow volume data. Specifically, we stack graph-structure-based layers based on GAT, recurrent layers based on LSTM and an output layer based on fully connected neural networks in the proposed model.

3. We conduct extensive experiments using a real-world traffic dataset. The results show that the hybrid GLM reduces the prediction error by approximately 6% to 10% as compared to the best baseline.
4. The motivation behind the hybrid GLM is to effectively and accurately predict the short-term metro passenger flow in cities that could help urban managers to improve traffic efficiency. Passenger flow prediction enables a variety of intelligent applications. It can help citizens to plan routes and schedule departure times. Moreover, the hybrid GLM brings new opportunities to artificial intelligence (AI) techniques on the construction of ITS, which is beneficial for building smart cities in a new era.

The rest of this paper is organized as follows. In Section 2, we review the related work, discuss the methods used, and their limitations. Section 3 describes the prediction problems and relevant theories. Section 4 introduces the structure of the hybrid GLM. Section 5 shows the case analysis and the corresponding results. Section 6 summarizes the achievements and limitations of this study, as well as future work.

2. Related Work

Passenger flow prediction is one of the major research issues in geo-information systems. Extensive research has been conducted to solve the problem. Existing methods can be categorized into three broad types: statistical methods, machine learning methods and deep learning methods.

2.1. Statistical Methods

Statistical methods predict future values based on previously observed values through time-series analysis [9]. As passenger flow data are a kind of time series data, it is feasible to use statistical methods to solve the prediction task. The statistical methods contain the autoregressive integrated moving average (ARIMA) model [13] and its variations [14,15], logistic regression model [16], Kalman filtering [17], etc. Liu [13] used a model based on ARIMA to forecast the rail traffic and found the results are superior than a back-propagation (BP) network. Ding [18] integrated ARIMA and GRACH models to forecast the subway short-term ridership accounting for dynamic volatility. Liang [19] combined the Kalman filtering and K-nearest neighbor (KNN) approach to handle different variation trends in the passenger flow data.

These methods can well capture the linear features but may neglect the non-linear features of passenger flow. However, passenger flow is often influenced by a variety of factors. The prediction performance of the statistical methods may worsen significantly if the data are not stable.

2.2. Machine Learning Methods

ML methods can map the complicated non-linear relations between input and output data, which can address the issue of statistical methods. Support vector machine (SVM) [20] is one of the most widely used ML models. It can strike a compromise between prediction accuracy and generalization ability based on the structural risk minimization principle [21]. Zhang [22] used the SVM to predict traffic flow, which obtained better results than linear regression models. Hybrid models based on SVR also have been widely used [23,24]. Li [25] integrated the seasonal autoregressive integrated moving average (SARIMA) model and SVM to establish a traffic flow prediction model. Cao [26] improved the parameter of the SVM model by using partial swarm optimization (PSO) for traffic flow prediction. Wang [21] proposed an SVM online model for capturing the periodicity and non-linear characteristics of short-term metro ridership, which extracts inputs feature via the SARIMA model and optimizes the parameters via PSO. Some other ML models, such as Bayesian networks, random forests (RFs), BP neural networks, KNN, etc., are also used in traffic flow prediction. Roos [27] proposed a Bayesian network for traffic flow forecasting, which can be used in incomplete data. Liu [28] combined RF methods to predict passenger flow, which puts more focus on input feature combination. Zhang [29] combined principal component analysis (PCA) and error BP networks to predict bus passenger flow, which increases the convergence speed. Bai [30] used enhanced KNN methods by considering the trend factor and time interval factor of passenger flow, which gets a better performance than the BP network and original KNN method.

ML methods greatly improved the accuracy of the metro passenger flow prediction. However, the performance of ML methods heavily depends on the manually designed features. Thus, it is hard to yield the best results for passenger flow prediction due to the complex and huge spatio-temporal data. Nowadays, it is rare to apply a single ML model to passenger flow prediction.

2.3. Deep Learning Methods

Due to the complexity of spatio-temporal data, most state-of-the-art literatures apply DL methods for passenger flow prediction. Compared with ML methods, DL methods can automatically extract essential features from raw data and make them robust with respect to variations in inputs [31].

RNNs [32] are good at dealing with complicated sequence information. As passenger flow is a kind of time series data, RNNs and their successors such as long short-term memory (LSTM) networks [33] and gated recurrent units (GRU) [34] are commonly used in the prediction task. Zhao [35] proposed a traffic forecasting model based on long short-term memory (LSTM) networks and achieved a good performance. Zhang [36] used a GRU-based method to predict urban traffic flow. Later, HAN [37] improved the optimizer of the LSTM, yielding better results than LSTM. Lin [38] used random forest (RF) to calculate the feature importance and applied the LSTM to predict metro passenger flow. However, the models mentioned above only capture the temporal dependency but neglect the spatial dependency, so that they cannot optimize the performance for the entire networks.

CNNs [39] are originally designed for data with regular grids, such as images. Some works apply CNN to identify spatial dependency with various localized filters or kernels. Zhang [40] input the spatio-temporal flow data into regular grids and proposed a model based on CNN called DeepST, which contains three fragments, denoting recent time, near history, and distant history. This is a milestone in passenger flow prediction. From then on, CNN-based models became prevailing in passenger flow prediction. Zhang [41] continued to integrate the residual unit to propose ST-ResNet. Residual networks can increase the model depth to capture characters with longer distances and more complex structures. Yu [42] designed a three-dimensional CNN network to achieve large-scale prediction on traffic flow. To capture the spatio-temporal dependency, scholars exploit RNNs in combination with CNN-based networks for passenger flow prediction. Ren [7] combined the ResNet and the LSTM to form a hybrid model, HIDLST, which yields better results than ST-ResNet. Qiao [6] utilized a one-dimensional CNN and LSTM for flow prediction.

However, CNN-based models must be conducted on regular grid data, which means it cannot capture the irregular topological relation of non-Euclidean data. The topological relation often exists in traffic infrastructures, such as stations, metro lines, roads, etc. Therefore, we should consider the topological information of traffic networks. With the development of graph theory [11,43,44], it is feasible to apply a graph structure to passenger flow prediction. Yu [45] used GCN for capturing the spatial relationship and applied one-dimensional CNN convolution to explore the temporal relationship. The proposed STGCN model was verified using the datasets of PEMS and Beijing. Zhao [46] proposed the temporal graph convolutional network (T-GCN) model, combining the graph convolutional network (GCN) with the GRU for traffic prediction. Zhang [47] utilized the GCN and 3D-CNN to model passenger flow. Ye [10] designed three spatial matrixes to extract the spatial dependency of a neighbor of different distances. However, because of the use of GCN, these models can only be used in asymmetric networks. Based on the attention mechanism, Guo [8] proposed a spatio-temporal convolutional network (ASTGCN) model, which can effectively capture the dynamic spatio-temporal dependency in traffic data. Zhang [48] combined ResNet, GCN, and attention LSTM to build a hybrid model, ResLSTM, which has achieved good results on the prediction of metro passenger flow in Beijing. However, these models neglect the improvement of the adjacent matrix.

2.4. Summary

Statistical methods and ML methods can be used for metro passenger flow prediction. However, both kinds of models cannot accept raw inputs for prediction. Therefore, it is hard to obtain higher prediction accuracy due to the complexity and randomness of the spatio-temporal data.

Among the DL methods, the graph-structure-based model is the hot spot regarding passenger flow prediction. Though the current literature shows progress in the given tasks, there are a number of knowledge gaps that need to be addressed, which are (1) GCN-based models require the network structure to be symmetric, so they are not applicable to asymmetric networks in a city; (2) most graph-structure-based methods only care about the effect of the adjacent nodes but ignore the nodes located a little further; and (3) some graph-structure-based models only capture the spatial dependency but ignore the temporal dependency and external factors.

Our model is inspired by GAT [12], which can model the topological spatial dependency of asymmetric networks. Aiming at solving the above problems, we proposed the hybrid GLM to extract spatio-temporal dependency and external dependency seamlessly.

3. Preliminary

3.1. Problem Definition

The research objective is to seamlessly model the spatial and temporal dependency of metro passenger flow data. The spatial dependency refers to the influence between metro stations, while the temporal dependency refers to the influence of historical metro passenger flow to the current time point. Moreover, metro passenger flow is affected by some external factors, such as weather conditions, temperature, holidays, air quality, etc. The external factors may affect human mobility, which increases the uncertainty of the metro passenger flow prediction. For example, people tend to stay at home rather than go out for dinner on a rainy day.

Thus, the problem of spatio-temporal metro passenger flow forecasting can be regard as an equation, Target = $F_{prediction}(I_S, I_T, I_E, W)$, where Target is the target passenger flow of time t, $F_{prediction}$ is the model used to tackle the problem, I_S presents the input related to the spatial dependency, I_T presents the input of the temporal dependency, I_E is the external factor influencing the metro passenger flow, and W denotes the parameters to be learned. The whole process of prediction is shown in Figure 1. Our goal is to use the historical passenger flow to predict the metro passenger flow at a certain moment.

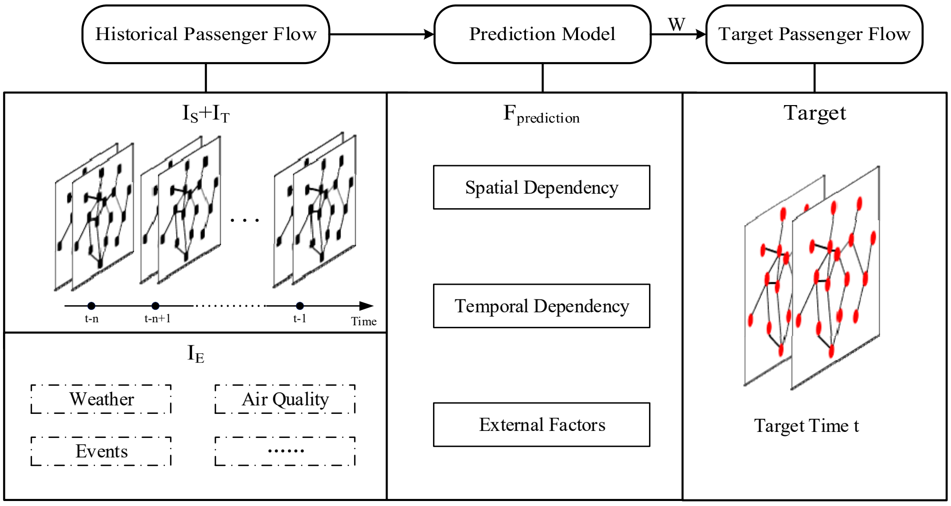


Figure 1. The whole prediction processes.

The input data come from three fragments of historical flow data, close, daily, and weekly patterns. The close pattern refers to the recent time. Daily and weekly patterns denote the historical passenger flow at the same time as the target time, but in daily or weekly periodicities [7]. If the target time interval is 7:00 a.m. to 7:10 a.m. on Saturday, the close pattern refers to the time close to 7:00 a.m., such as 6:50 a.m., 6:40 a.m., 6:30 a.m., etc. The daily pattern refers to 7:00–7:10 a.m. every day for the prior d days. The weekly pattern is the time from 7:00–7:10 a.m. every Saturday of the prior w weeks.

The input data of close, daily, and weekly patterns can be described as Equations (1)–(3).

$$I_C = (I_{t-1}, I_{t-2}, \ldots, I_{t-c}), \tag{1}$$

$$I_D = (I_{t-1\cdot n}, I_{t-2\cdot n}, \ldots, I_{t-d\cdot n}), \tag{2}$$

$$I_W = (I_{t-1\cdot 7\cdot n}, I_{t-2\cdot 7\cdot n}, \ldots, I_{t-w\cdot 7\cdot n}), \tag{3}$$

where I_t represents the passenger flow of the target time t. I_C, I_D and I_W denote the historical passenger flow of the close, daily, and weekly patterns respectively. Assume that the number of time intervals in I_C, I_D, and I_W are c, d, and w. Besides, the total number of a day's time interval is n.

3.2. Principle of GAT

Graph attention networks (GATs) were proposed by Petar Veličković [12], which can be operated on graph-structured data. GAT introduces an attention mechanism into the graph structure and applies masked self-attentional layers that can assign different importance to different nodes within a neighborhood without costly matrix operations [49]. Besides, a graph structure is injected into the model as a mask. In this way, neither the matrix operation nor the entire graph structure is needed. Therefore, we can apply GAT to the incomplete graph, directed graphs, asymmetric graphs, and dynamic graphs. The graph structure is one of the typical organizations of citywide metro data. Therefore, we can apply GAT to predict citywide metro passenger flow.

Using GAT, the first step is to construct a graph structure $G(V, E)$, where V is the node, namely the metro station; and E is the connection line, namely the metro line between two neighbor stations. Then we need to build a block layer to construct the graph attention networks (by stacking this layer). The inputs are the features of the nodes, which refer to the passenger flow of each metro station. It can be described as $h = \{\vec{h}_1, \vec{h}_2, \ldots, \vec{h}_N\}, \vec{h}_i \in R^F$, where N is the number of nodes and F is the feature of each node. The layer produces a set of new features $h' = \{\vec{h}'_1, \vec{h}'_2, \ldots, \vec{h}'_N\}$, $\vec{h}'_i \in R^{F'}$ as outputs, where F' represents the output features. A shared linear transformation is applied to each node, which is parametrized by a weight matrix, $W \in R^{F' \times F}$. Then we perform a shared self-attention mechanism on the nodes, which is denoted as $a : R^{F'} \times R^{F'} \to R$. Its objective is to compute attention coefficients $e_{ij} = a(W\vec{h}_i, W\vec{h}_j)$, where e_{ij} shows the importance of j to i. Later, we should inject a mask based on the graph structure, which means we only do calculations on $j \in N_i$, where N_i is the neighborhood of node i (as shown in Figure 2, we only calculate the influence of Node 1, 7, 5, 6 on Node 3). Lastly, we can apply LeakyRelu for calculation. The attention coefficient can be shown as follows:

$$\alpha_{ij} = \frac{\exp(LeakyReLu(\vec{a}^T[W\vec{h}_i||W\vec{h}_j]))}{\sum_{k \in N_i} \exp(LeakyReLu(\vec{a}^T[W\vec{h}_i||W\vec{h}_j]))}, \tag{4}$$

where $^{\mathrm{T}}$ represents transportation and $||$ is the concatenation operation.

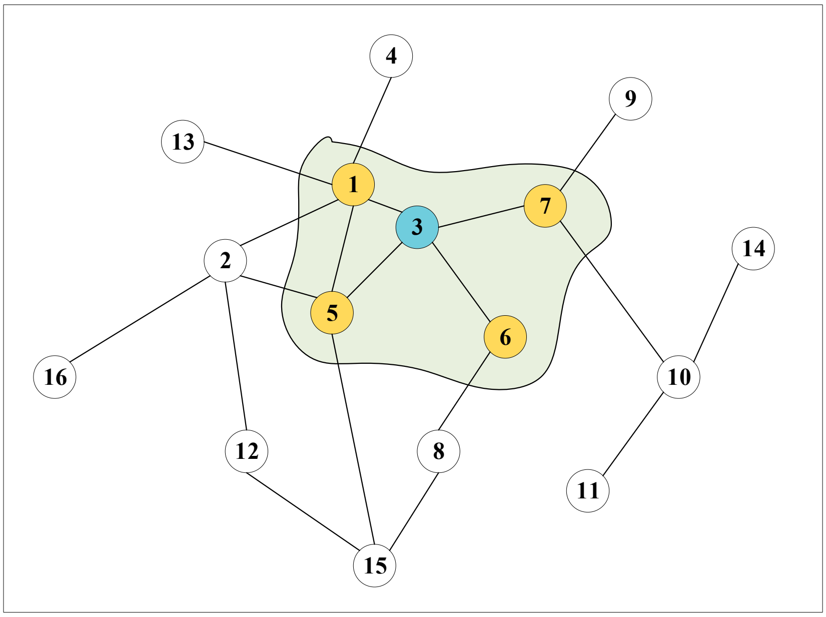


Figure 2. Neighborhoods of Node 3.

To stabilize the learning process of self-attention, the author extended their mechanism to employ multi-head attention. The features are connected, and the output of feature character is as follows,

$$\vec{h}'_i = \overset{K}{\underset{k=1}{\Big\|}} \sigma\Big(\sum_{j\in N_i} \alpha_{ij}^{k} W^{k} \vec{h}_j\Big), \tag{5}$$

where $||$ represents concatenation, α_{ij}^{k} are the normalized attention coefficients computed by the k-th attention mechanism, and W^k is the corresponding input linear transformation's weight matrix [12]. In particular, if we perform multi-head attention on the final prediction layer of the network, we will employ averaging on the layers rather than concatenation. The process is shown in Figure 3, which contains three heads. We aim to calculate the graph attention coefficient of Node 2,3, and 4 to Node 1 in the picture.

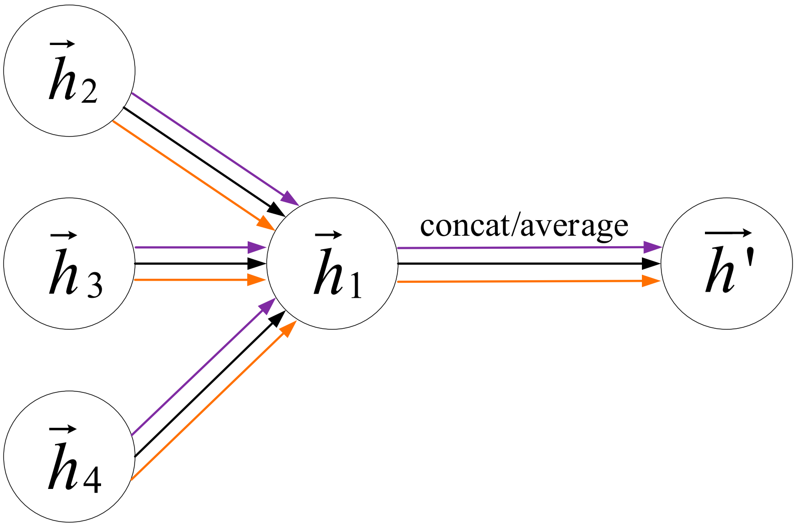


Figure 3. Multi-head attention (K = 3) for Node 1 on its neighborhood.

We calculate the attention coefficients of all neighbor nodes and add them to obtain the final attention coefficient of the node. In this way, the influence between nodes can be captured and the spatial dependency of passenger flow can be accurately obtained.

3.3. Principle of LSTM

Long short-term memory (LSTM) networks were proposed by Hochreiter and Schmidhuber [50]. It is a kind of recurrent neural network (RNN). The objective of LSTM is to model the longer-distance features of the time series. LSTM can tackle the problems of gradient exploding and vanishing in the traditional RNNs. The LSTM consists of three parts, an input layer, a recurrent hidden layer, and an output layer. Different from the traditional RNNs, the recurrent hidden layer of the LSTM contains a special memory block, whose core structure is shown in Figure 4. The memory block contains memory cells with self-connection that store the temporal state of the network at each time step [51]. The temporal state is controlled by three gates: the forget gate, the input gate, and the output gate. The input gate is to protect the memory contents from irrelevant inputs. The forget gate is to forget some useless message. The output gate is to export the outputs.

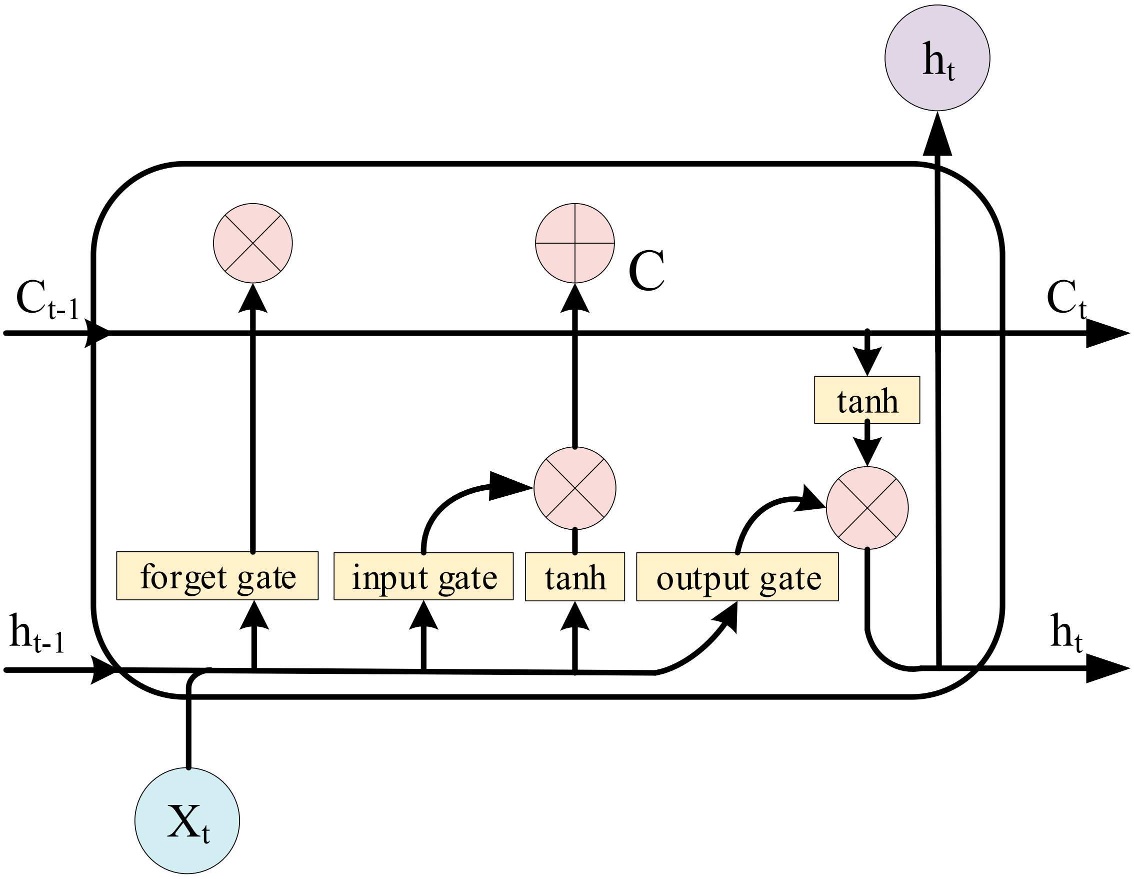


Figure 4. The core unit of the long short-term memory (LSTM) network.

In Figure 4, X_t is the input of the current time point, h_t is the output of the hidden layer, h_{t-1} is the output hidden layer of the previous time interval, C is the input state of the cell, C_t is the output state of the cell, and C_{t-1} is the output state of the previous time interval. The coefficients of input gate, the forget gate, and the output gate in LSTM can be calculated in Equations (6)–(8) below.

input gate:

$$i_t = \sigma(W_{xi} \cdot \mathrm{X}_t + W_{hi} \cdot h_{t-1} + b_i), \tag{6}$$

forget gate:

$$f_t = \sigma(W_{xf} \cdot \mathrm{X}_t + W_{hf} \cdot h_{t-1} + b_f), \tag{7}$$

output gate:

$$o_t = \sigma(W_{xo} \cdot \mathrm{X}_t + W_{ho} \cdot h_{t-1} + b_o), \tag{8}$$

where W_{xi}, W_{xf}, and W_{xo} are learnable weight parameters connecting X_t with the input gate, forget gate, and output gate. W_{hi}, W_{hf}, and W_{ho} are weight parameters connecting h_{t-1} with three gates. b_i, b_f, and b_o are learnable offset parameters. σ is the sigmoid function: $\frac{1}{1+\exp(-x)}$.

The input state of the cell is as follows:

$$C = \tanh(W_{xC} \cdot \mathrm{X}_t + W_{hC} \cdot h_{t-1} + b_C), \tag{9}$$

where W_{xC} is a weight parameter connecting X_t with the inputs, W_{hC} is the parameter matrix connecting h_{t-1} with the cell inputs, b_C is the learning offset parameters, and tanh is the tangent function.

The output state of the cell is as follows:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot C, \tag{10}$$

where i_t, f_t, C_t, C_{t-1}, and C share the same dimension.

The output of the hidden layer is as follows:

$$h_t = o_t \cdot \tanh(C_t). \tag{11}$$

In short, the LSTM can "remember" the needed information and "forget" the useless information. Thus, the LSTM owns the strong ability to process a time series with a longer temporal dependency. Applying the LSTM to the metro passenger flow prediction can capture the temporal dependency of the data, which contributes to the accuracy of the prediction model.

4. Model Development

Citywide metro passenger flow prediction is a typical spatio-temporal modeling problem. Therefore, we propose a model combining GAT and LSTM, which is called the hybrid GLM. Besides, the proposed model consists of a multi-time pattern. With GAT, the hybrid model can deal with topological problems better than the other models. The model consists of five parts, Branches 1–5. **Branches 1–3** use the GAT structure to capture the spatial dependency in the close, daily, and weekly patterns. **Branch 4** uses the LSTM to capture the temporal dependency through the fused close, daily, and weekly patterns. **Branch 5** shows the impact of external factors. Moreover, an LSTM layer is used to obtain the output data. The detailed model architecture is presented in Figure 5.

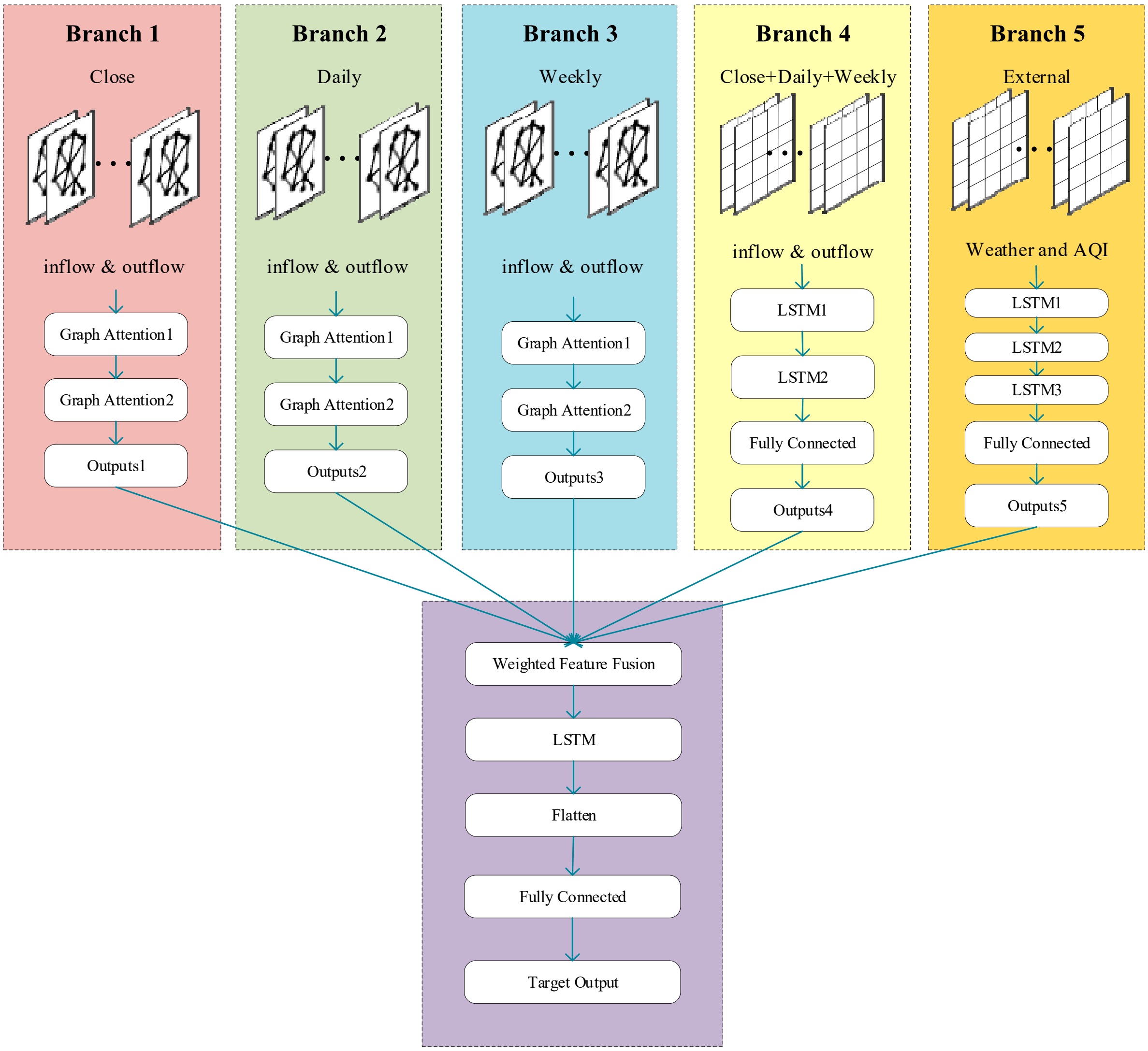


Figure 5. The architecture of the hybrid GAT and LSTM Model (hybrid GLM).

4.1. Branches 1–3: Spatial Dependency

The influence of historical passenger flow can be divided into three patterns: close pattern, daily pattern, and weekly pattern. We take the three patterns as three parts, which are sent into the GAT for training. We take the three patterns apart for two reasons. On one hand, if we regard three parts as one input, the data in the GAT may be in a great amount. Thus, the training process may be very slow. On the other hand, the spatial correlation among the three patterns is not strong, so there is no need to train them together. The GAT structure in our model can capture the topological characters of the passenger flow, and it can be used in asymmetric networks. Every GAT structure in Branches 1 to 3 contains two graph attention layers. The topological relationship between the nodes are used to construct the adjacency matrix which is regarded as a layer mask. We utilize the mask to capture the topological relations between metro stations. To observe the correlations of further metro stations, we improved the traditional adjacent matrix. In the traditional adjacent matrix, we put 1 in the matrix if the two nodes can be connected by lines. However, we want to capture the spatial correlation of some nodes located a little further for better predicting. To that end, we set 4,3,2, and 1 as the weight of the closest nodes, less close, much less close, and further nodes, respectively. Besides, if the nodes are connected by several edges, we add 0.5 to the weights per edge. Doing so can also facilitate large-scale metro network prediction.

4.2. Branch 4: Temporal Dependency

Another obvious character of metro passenger flow is its temporal dependency, which refers to the impact of the historical passenger flow on the current time point. There are three obvious aspects of temporal correlation: proximity, trend, and periodicity. Proximity means the influence of the closest time intervals. Trend means the overall trend over a period of time. Periodicity is the influence of a longer time. In our model, the time intervals of the close, daily, and weekly patterns are merged and sent into the LSTM. LSTM is a special RNN that uses gate structures to determine the necessity of the information. LSTM solves the gradient explosion and the gradient disappearance problems of traditional RNNs, making it available to capture the characters of much longer temporal distance. In the hybrid GLM, a two-layer LSTM is used for the metro passenger flow sequence. Then, the data are flattened and fully connected with 578 neurons. Through Branch 4, the temporal dependency can be obtained and the overall periodicity of the metro passenger flow can be studied.

4.3. Branch 5: External Influence

Apart from the spatial and temporal factors, some external factors may affect the prediction of metro passenger flow. For example, people tend to stay at home on sandstorms or heavily polluted days. Major events, such as the National Day, may make the metro passenger flow reach a new peak. External factors are essential references for people to schedule their travel plans. At present, only a few models introduce external factors into the prediction model and they pay little attention to air quality. Our model selects 11 external factors, which can be divided into three categories: weather conditions (maximum temperature, minimum temperature, and rainy day or not), air quality (AQI, PM2.5, PM10, NO_2, CO, O_3, and SO_2), and events (whether the day is a holiday or not). We use the time series of these factors to analyze the external influence. The external data are recorded every hour, and some examples are shown in Table 1.

Table 1. Examples of external factors.

Date/Time	Highest (°C)	Lowest (°C)	Holiday	RainyDay	AQI	PM2.5	PM10	SO_2	NO_2	O_3	CO
1 April 2015/6:00	24	17	No	Yes	113.11	85.00	94.67	18.22	85.78	12.33	1.44
1 April 2015/7:00	24	17	No	Yes	115.44	87.00	101.44	19.33	88.11	14.00	1.57
1 April 2015/8:00	24	17	No	No	124.89	94.44	105.89	23.67	97.00	17.44	1.44
1 April 2015/9:00	24	17	No	No	116.11	87.56	82.89	26.11	83.33	32.11	1.20
1 April 2015/10:00	24	17	No	Yes	84.33	62.11	58.89	25.67	74.11	50.33	1.02

Note that the external data are recorded every hour. However, the time interval is 10 min in our experiment, which means a day contains 144 time intervals and an hour contains 6 time intervals. Therefore, the 6 intervals share the same recorded data in an hour. For example, the weather condition data from 6:00 to 6:10 will share the recorded data from 6:00 to 7:00, as shown in the first row of Table 1.

We normalized the external data so that all the quantities are in the same range. We performed one-hot encoding on the Boolean values like Holiday and RainyDay. Then the processed data were sent into the stacked LSTM layers. We built a three-layer LSTM and each layer has 256 neurons. At last, the output of Branch 5 and the outputs of the prior 4 Branches were put into the feature fusion part for training.

4.4. Feature Fusion

Because the output data from the five branches are in identical shape, we can fuse the five parts. However, the influence of the different parts varies. We adopt the parametric-matrix-based method, whose function is shown below.

$$Fusion = \sigma(W_1 \circ O_1 + W_2 \circ O_2 + W_3 \circ O_3 + W_4 \circ O_4 + W_5 \circ O_5), \tag{12}$$

where *Fusion* is the prediction target after fusion, $\circ$ presents the Hadamard product, O_1, O_2, O_3, O_4, and O_5 are the outputs from the five branches, and W_1, W_2, W_3, W_4, and W_5 are the corresponding learnable weights. Then the results after fusion are activated by the activation function σ (i.e., ReLU). An LSTM layer with 64 neurons was applied after feature fusion. The LSTM output was subsequently flattened and fully connected with 578 neurons to generate the final outputs.

4.5. Model Training

To train the hybrid GLM, the mean-square error (MSE) is used as the loss function. As shown in Equation (13), y_i is the available ground value, $\hat{y}_i$ is the predicted value and n is the number of samples. The original data were divided into three parts, a training dataset: a validating dataset, and a testing dataset. We use the training dataset for training in batches and the loss will be calculated per batch. Besides, we apply Adam as an optimizer when back propagation training. Adam is generally regarded as being fairly robust to the choice of hyperparameters, though the learning rate sometimes needs to be changed from the suggested default [52] (pp. 309). After minimizing the loss, all trainable parameters are trained.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{13}$$

5. Case Study

5.1. Experiment Data

The metro passenger flow data used in this study were collected from Smart Card Data (SCD) of the metro system of Shanghai, China. The study area and the corresponding metro lines are shown in Figure 6. The time span of the data is between April 1st and April 30th in 2015. During this period, there were about nine million card records per day, covering 289 metro stations. Parts of the original data are shown in Table 2. The corresponding field descriptions are shown in Table 3.

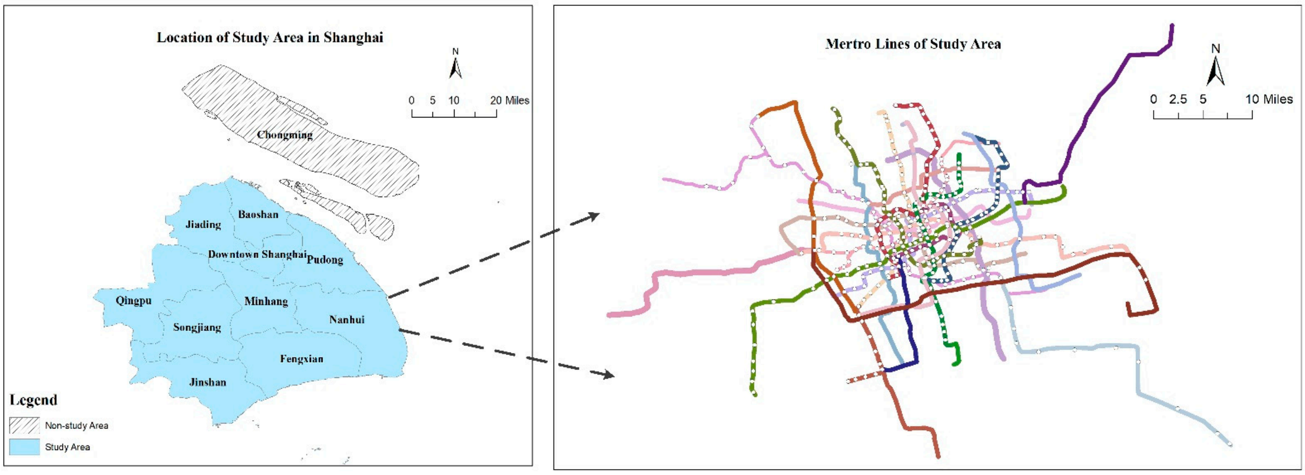


Figure 6. Location of the study area and the metro lines of Shanghai.

Table 2. Examples of the original Smart Card Data (SCD).

Cardnum	Date	Time	Linename	Business	Figure	Attribute
602141128	2015-04-01	09:07:57	No.11 East Changji Road	subway	6.0	no discounts
2201252167	2015-04-01	19:20:33	No.7 Changzhong Road	subway	4.0	no discounts
2201252167	2015-04-01	08:55:44	No.1 South Shanxi Road	subway	4.0	no discounts
2201252167	2015-04-01	18:43:14	No.1 South Shanxi Road	subway	0.0	no discounts
2201252167	2015-04-01	08:19:00	No.7 Shangda Road	subway	0.0	no discounts
602141128	2015-04-01	09:07:57	No.11 East Changji Road	subway	6.0	no discounts

Table 3. Data structure of the SCD.

Field Name	Description	Field Type
Cardnum	Unique number for each card	varchar
Date	Detailed date of transaction	datetime
Time	Detailed time of transaction	datetime
Linename	Unique number of metro and name of metro station	varchar
Business	Travel way of trip	varchar
Figure	Price of trip	float
Attribute	Discount or not	varchar

According to the life experience, we can obtain the metro passenger inflow and outflow from the original data. The field Figure is the key to tell whether the flow is inflow or outflow. Take Row 1 in Table 2 as an example, which is a record of passenger outflow. Apparently, a Figure of zero represents inflow, while a Figure that is non-zero is outflow.

Then we put forward a definition of time interval (TI) for counting passenger flow. We choose 10 min, 15 min, 20 min and 30 min as the TIs, respectively. We should count the passenger flow every 10 min if the TI is 10 min. Then a day has 144 time slices. However, we chose 6:40 a.m. to 11:00 p.m. as the studying period according to human activities. Therefore, there were 98 time slices in a day.

The training set includes observations from 1–23 April 2015; the validation set is from 24–25 April 2015 (including one working day and one non-working day). We selected the last five days, 26–30 April 2015, as the testing period, which contains four working days and one non-working day.

5.2. Evaluation Metrics

To measure the performance of the different flow prediction models, we chose root-mean-square error (RMSE) and mean absolute percentage error (MAPE) as the evaluation metrics. They are calculated through prediction values and available ground values. Definitions of the two metrics are shown in Equations (14) and (15). From the definition, we can know that the smaller the value is, the better the model performs.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}, \tag{14}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|, \tag{15}$$

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

5.3. Environment and Training Settings

Experiments were mainly run on a GPU platform with an NVIDIA GeForce GTX1050 Ti graphics card, whose detailed information is shown in Table 4. Python libraries, including scikit-learn, Keras, and TensorFlow were used to build our model.

Table 4. Experimental environment.

Items	Parameters
OS	Windows 10
Memory	16GB
CPU	Inter®Core(TM) i5-8500 CPU @3.00GHz
GPU	NVIDIA GeoForce GTX 1050Ti
CUDA version	10.2
cuDNN version	10.2
Keras version	2.1.5
TensorFlow version	1.14.0
scikit-learn version	0.23.2

The procedure of tuning the parameters is important for DL prediction. Here, only the final settings are listed, which were proven to be the optimal parameters. The detailed tuning procedure will be presented in Section 5.6. In our experiments, the number of time intervals for close, daily, and weekly patterns were set as c = 7, d = 1, and w = 1. For Branches 1–3, we stacked two graph attention layers for every branch. The first layer had 6 output neurons, while the second layer had 2. We set the attention head as 12 for better training. To avoid overfitting, the dropout layers were added between the two graph attention layers. The dropout rate was set as 0.6. For Branch 4, we stacked two LSTM layers with 600 neurons each. A fully connected layer consisting of 578 neurons was applied in the end. To capture the influence of the external factors, we utilized a 3-layer LSTM with 256 neurons. For the feature fusion part, an LSTM layer and a fully connected layer consisting of 64 and 578 neurons were applied, respectively.

We trained the hybrid GLM model by minimizing the MSE for 200 epochs with a batch size of seven. We also used the Early Stopping techniques to avoid overfitting. The initial learning rate was set at 5×10^{-4}, with a decay rate of 0.95 after every 20 epochs. The training loss and validation loss become stable after 200 epochs, which shows the robustness of the proposed model.

5.4. Baseline Models

We compare the hybrid GLM with the following 10 baseline models (including one statistical method, two ML methods, and seven DL methods) to evaluate the performance. To make a fair comparison, all these models take the close, daily, and weekly data as inputs. The Adam optimizer is used for all the models. The descriptions of these baseline models are as follows. Related abbreviations of the baseline models are shown in Table 5.

Table 5. Abbreviation table of baseline models.

Abbreviation	Item	Abbreviation	Item
LR	**L**inear **R**egression	ResNet	**Net**works with **Res**Unit
KNN	**K**-**N**earest **N**eighbor	STGCN	**S**patio-emporal **G**raph **C**onvention **N**etworks
RSVR	**S**upport **V**ector **R**egression with **R**adial-basis Function	GAT	**G**raph **A**ttention **N**etworks
LSTM	**L**ong **S**hort-**T**erm-**M**emory	GLM_NoE	**G**AT and **L**STM **M**odel without **E**xternal Factors
CNN	**C**onvolutional **N**eural **N**etwork	GLM_NoIA	**G**AT and **L**STM **M**odel without **I**mproved **A**djacent Matrix

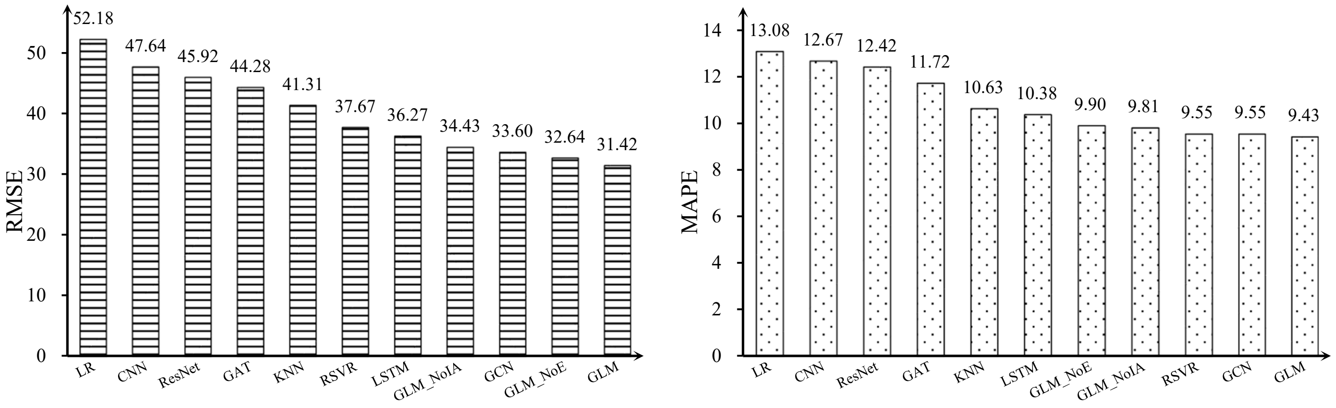
- **LR:** Linear regression (LR) [53] is a statistical model that is used to capture the relationship between a response and one more explanatory variables. Based on this, it is employed to perform future predictions [54];

- **KNN:** K-nearest neighbor (KNN) regression [55] is a commonly used method in nonparametric regression. We also employ PCA to select the principal components before inputting the data into KNN;
- **RSVR:** A typical machine learning method [20]. The kernel of SVR in scikit-learn is set as a radial-basis function (RSVR);
- **LSTM:** Long short-term-memory (LSTM) networks [50]. LSTM is a special kind of RNNs, which is capable of learning long-term temporal dependencies. The model consists of two stacked LSTM layers and one fully-connected layer;
- **CNN:** A convolutional neural network (CNN) [56], which transforms the metro-network-based passenger flow into a two-dimensional image. The vertical axis represents the metro stations, the horizontal axis represents time;
- **ResNet:** A model combined with CNN and ResUnit (ResNet) [41]. It was used in the traffic field once. However, we do not embed the external factors in our study;
- **STGCN:** A model that generalizes CNNs to non-Euclidean data, which is used in the spectral domain with graph Fourier transforms. In our study, we utilize the spatio-temporal graph conventional networks (STGCN) proposed by Han [57] as a baseline model;
- **GAT:** Graph attention networks (GAT) [12]. GAT is a kind of graph neural networks, which can analyze the topological relations of nodes. Two graph attention layers are used in the model;
- **GLM_NoE:** We delete Branch 5 in the hybrid GLM;
- **GLM_NoIA:** We only use the traditional adjacent matrix as a mask layer compared with the hybrid GLM.

5.5. Results and Discussion

5.5.1. Different Networks Prediction Performance

Similar to the hybrid GLM, we tuned the hyperparameters for the other 10 baseline models and recorded the optimal hyperparameters. The final results are shown in Table 6. The MAPEs are calculated from the data whose actual value is not zero. In Table 6, FN_{CNN} presents the number of hidden neurons of one CNN layer. D_{CNN} refers to the number of hidden layers in CNN. F and D have similar meanings in other baseline models. L_C, L_D, and L_W present the length of the close, daily, and weekly patterns, respectively. K means the number of attention heads. $Kernel_{SVR}$ refers to the type of kernel in the SVM algorithm. Neighbor means the number of nodes that one class contains. K_s is the kernel size of the graph convolution. To further observe the prediction performance in a more intuitive way, we draw the bar pictures of the RMSEs and the MAPEs for all models in Figure 7.



(**a**) RMSE of different models.

(**b**) MAPE of different models.

Figure 7. The RMSE and MAPE rankings of the different models.

Table 6. The optimal hyperparameters of the baseline models.

No.	Model	Optimal Hyperparameters	RMSE	MAPE (%)
1	LR	$FN_{Dense} = 578$	52.18	13.08
2	CNN	$FN_{CNN1} = 256$, $FN_{CNN2} = 128$, $FN_{CNN3} = 64$, $FN_{CNN4} = 4$, $D_{CNN\,=}$ 4, $L_C = 7$, $L_D = 1$, $L_W = 1$	47.64	12.67
3	ResNet	$FN_{ResNet} = 128$, $D_{ResNet} = 2$, $L_C = 7$, $L_D = 1$, $L_W = 1$	45.92	12.42
4	GAT	$FN_{GAT1} = 6$, $FN_{GAT1} = 2$, $D_{GAT} = 2$, $K = 12$, $L_C = 7$, $L_D = 1$, $L_W = 1$	44.28	11.72
5	KNN	Neighbors = 5, $L_C = 7$, $L_D = 1$, $L_W = 1$	40.55	10.12
6	RSVR	$Kernel_{SVR}$ = rbf, $L_C = 7$, $L_D = 1$, $L_W = 1$	37.67	9.55
7	LSTM	$FN_{LSTM1} = 600$, $FN_{LSTM2} = 600$, $D_{LSTM} = 2$, $L_C = 7$, $L_D = 1$, $L_W = 1$	36.27	10.38
8	STGCN	$FN_{GCN} = 578$, $D_{GCN} = 5$, $K_s = 3$, $L_C = 7$, $L_D = 1$, $L_W = 1$	33.60	9.55
9	GLM_NoIA	$FN_{GAT1} = 6$, $FN_{GAT1} = 2$, $D_{GAT} = 2$, $K = 12$, $FN_{LSTM1} = 600$, $FN_{LSTM2} = 600$, $D_{LSTM} = 2$, $L_C = 7$, $L_D = 1$, $L_W = 1$	34.43	9.81
10	GLM_NoE	$FN_{GAT1} = 6$, $FN_{GAT1} = 2$, $D_{GAT} = 2$, $K = 12$, $FN_{LSTM1} = 600$, $FN_{LSTM2} = 600$, $D_{LSTM} = 2$, $L_C = 7$, $L_D = 1$, $L_W =1$	32.64	9.90
11	GLM	$FN_{GAT1} = 6$, $FN_{GAT1} = 2$, $D_{GAT} = 2$, $K = 12$, $FN_{LSTM1} = 600$, $FN_{LSTM2} = 600$, $D_{LSTM} = 2$, $L_C = 7$, $L_D = 1$, $L_W = 1$	31.42	9.43

As shown in Table 6 and Figure 7, the hybrid GLM outperforms most mainstream methods on Shanghai metro data with the smallest RMSR and MAPE. Compared with CNN, ResNet, and GAT, the hybrid GLM exhibits an obvious reduction in RMSE and MAPE. Specifically, compared with CNN, the hybrid GLM has a 34.11% relative reduction in RMSE and a 25.57% relative reduction in MAPE. Compared with ResNet, the hybrid GLM has a 31.58% relative reduction in RMSE and a 24.07% relative reduction in MAPE. Compared with GAT, the hybrid GLM has a 29.04% relative reduction in RMSE and a 19.54% relative reduction in MAPE. In addition, the hybrid GLM also outperforms LSTM. Compared with the LSTM, the hybrid GLM exhibits an RMSE reduction of 13.37% and a MAPE reduction of 9.15%. We regard CNN, ResNet, and GAT as Group1, and LSTM as Group2. The reason why the hybrid GLM is superior to Group1 and Group2 is that either Group1 or Group2 only captures spatial or temporal dependency. However, the hybrid GML combines the advantage of the GAT and LSTM for capturing spatio-temporal dependency.

Next, we compared the performance of the different models, the statistical models, the ML models, and the DL models. The statistical model, LR, performs worse than the ML models and the DL models. Among the DL models, we find that most models concerning spatial dependency get worse results than the models based on recurrent neural networks. Take CNN and LSTM as an example: LSTM exhibits an RMSE reduction of 23.87% and a MAPE reduction of 18.07% compared with CNN. The reason may be that it is more difficult to capture the spatial dependency than temporal dependency for citywide metro network prediction. As for the ML methods, KNN and RSVR, they perform worse than the DL model concerning temporal dependency, such as LSTM. However, they get better results than the DL models concerning spatial dependency, such as CNN, ResNet, and GAT. Compared with KNN, the hybrid GLM has a 23.94% relative reduction in RMSE and an 11.29% relative reduction in MAPE. Moreover, the hybrid GLM exhibits a reduction of 16.59% in RMSE and a reduction of 1.3% in MAPE compared with RSVR. Generally speaking, the hybrid GLM outperforms ML models in a sense.

Compared with the raster-based models, like CNN and ResNet, the graph-structure-based models have a smaller RMSE and MAPE. Compared with ResNet, the STGCN has a 26.83% relative reduction in RMSE and a 23.11% relative reduction in MAPE. Compared with CNN, GAT has a 7.05% relative reduction in RMSE and a 7.50% relative reduction in MAPE. Generally speaking, the RMSEs of the raster-based model are larger than 40, while the hybrid GLM only gets 31.42. The MAPEs of the raster-based model are larger than 12, while the hybrid GLM model only gets 9.43. From the results, we conclude that the graph-structure based model can capture the irregular spatial dependency better than the raster-based models for the citywide metro passenger flow prediction.

Then, we compared the hybrid GLM with STGCN and found that the hybrid GLM performs a little better. The hybrid GLM has a 6.49% relative reduction in RMSE and a 10.87% relative reduction in MAPE mainly because the GAT component in the hybrid GLM improves the original adjacent matrix to compensate for the asymmetric matrix problem of STGCN.

Lastly, we discuss the contribution of improved adjacent matrix and external factors. As shown in Table 6, the hybrid GLM has an 8.74% relative reduction in RMSE and a 3.06% relative reduction in MAPE compared with GLM_NoIA, which indicates the benefits of the improved adjacent matrix. Moreover, the hybrid GLM embeds external factors, which improves the model a little. We can learn from the results that the hybrid GLM exhibits an RMSE reduction of 3.74% and an MAPE reduction of 3.94% compared with GLM_NoE.

In summary, by integrating GAT and LSTM, the hybrid GLM can capture the spatio-temporal dependency better than several existing models.

5.5.2. Prediction Results of a Specific Metro Station

According to the results of the RMSE and MAPE of the different models, we selected four models that performed relatively well, namely, STGCN, RSVR, LSTM, and GLM. We are going to discuss these models in detail. We chose the People's Square metro station as an example, for it is one of the most crowded stations in Shanghai and it is located in the center of the city.

Figures 8 and 9 are the prediction results of passenger inflow and outflow per 10 minutes in People's Square metro station during the 4 working days and 1 non-working day. We set 6:40 a.m. to 11:00 p.m. as the time span. We can easily find the periodicity of metro passenger flow on the working day in Figures 8 and 9. The People's Square station is a typical working area. Therefore, the rush hours come at 5:00 p.m. every working day (the time interval is around 161, 271, 361, and 455 in Figure 8), which is the time to get off work. As for outflow volume, the rush hours come at 9:00 a.m. every working day (the time interval is around 113, 209, 308, and 410 in Figure 9), which is an office hour. As for the performance, the predicted values of the hybrid GLM, RSVR, LSTM, and STGCN models correspond well with the actual values. More specifically, in the red boxes of Figure 8, we can find the hybrid GLM catches the characters of the inflow volume in rush hours more accurately than the other three models. However, the hybrid GLM performs slightly worse in predicting the outflow volume in non-rush hours, which can be seen in Figure 9.

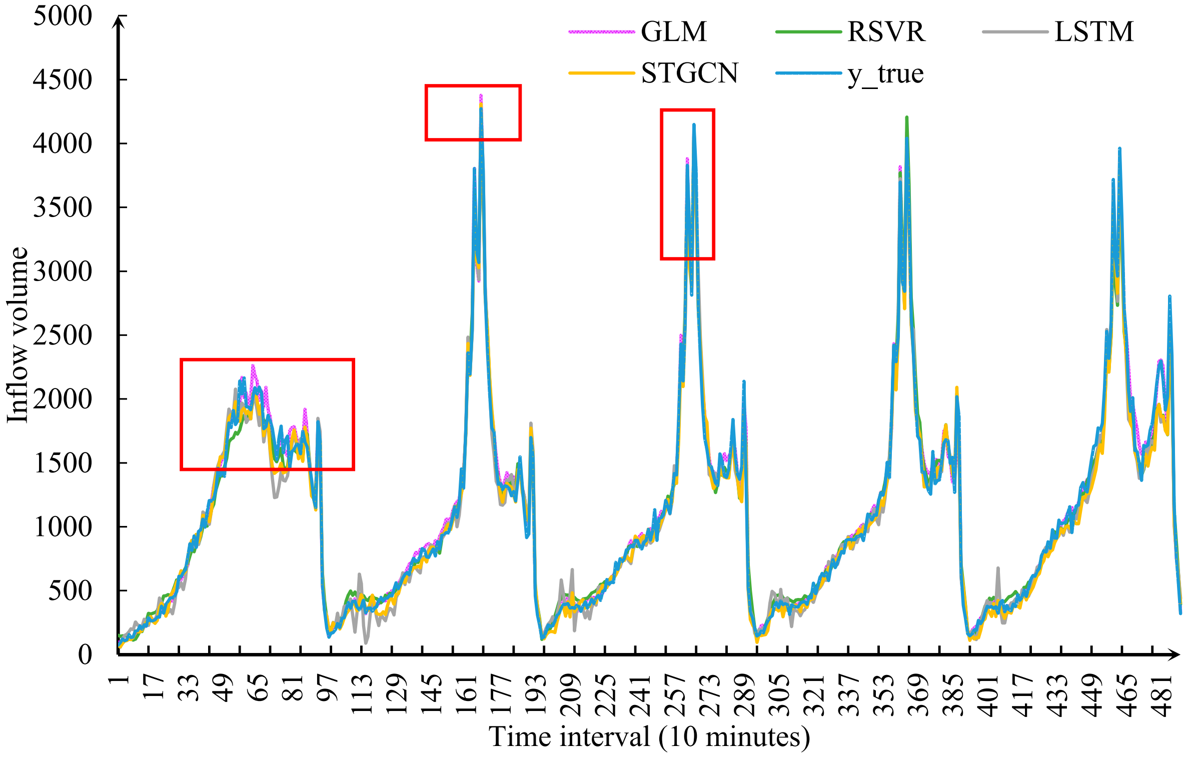


Figure 8. The prediction performance of inflow volume in the People's Square metro station. The time interval is 10 min and the test time span covers 1 non-working day and 4 working days.

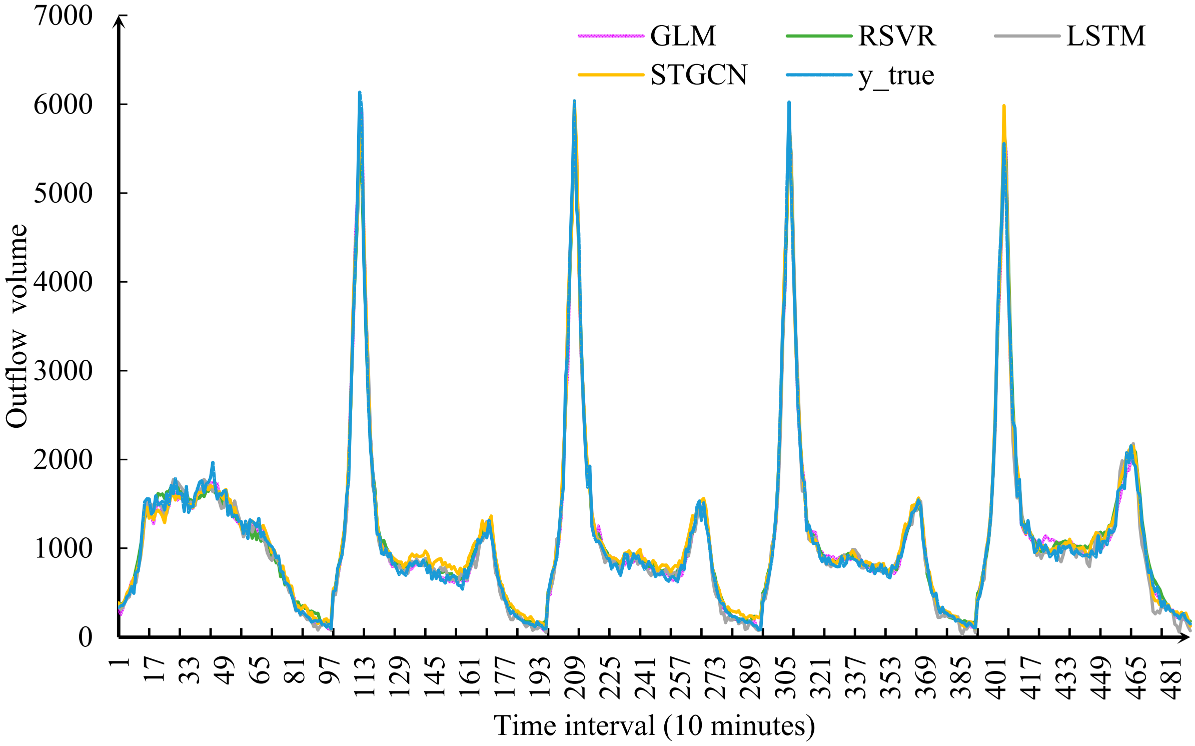
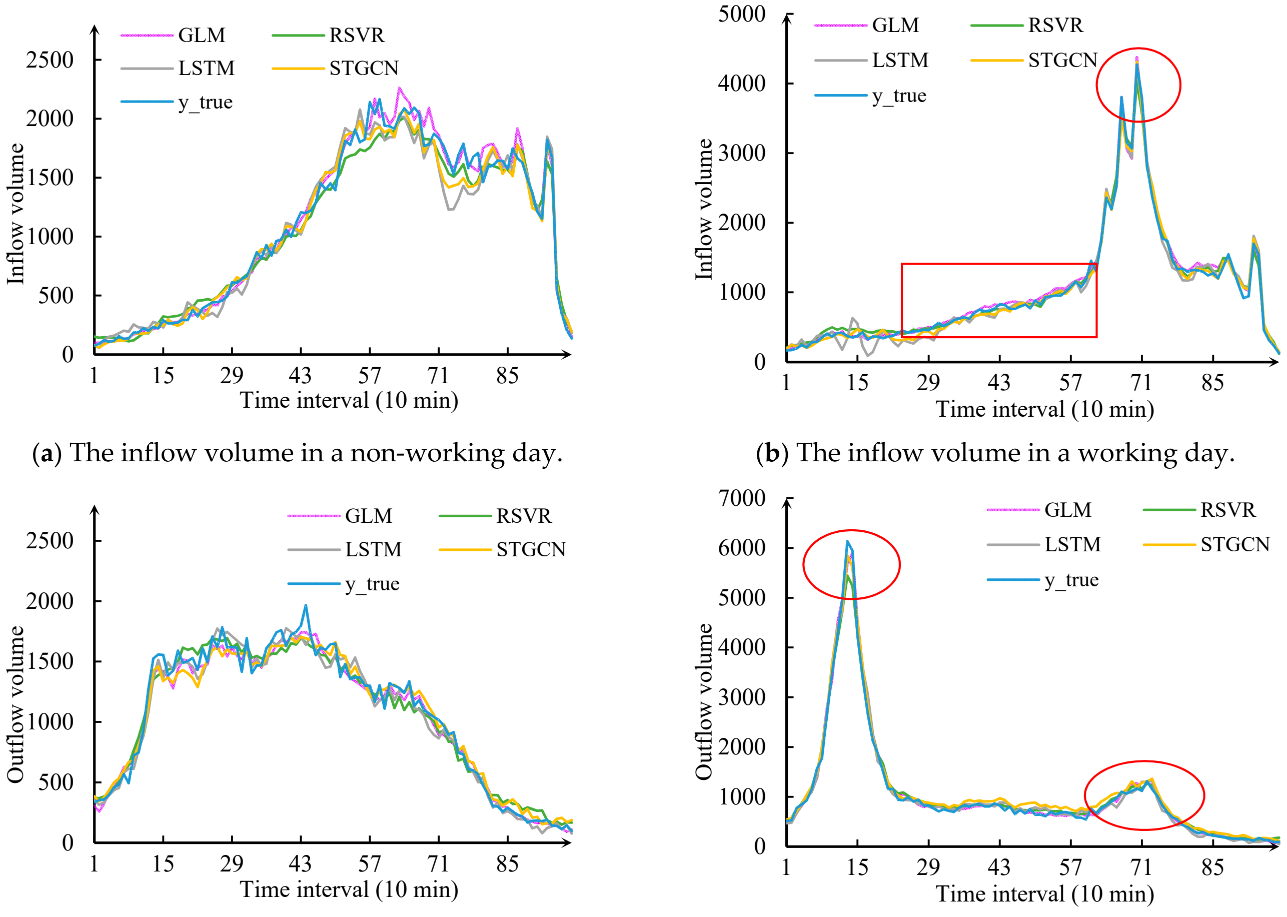


Figure 9. The prediction performance of outflow volume in the People's Square metro station. The time interval is 10 min and the test time span covers 1 non-working day and 4 working days.

In Figure 10, we draw the prediction results of non-workdays and working days specifically. There are 98 time slices in a day. As shown in Figure 10a,c, the passenger flow volume of the non-working days is more variable and the prediction results are therefore slightly worse. Comparing the overall predictive abilities of non-working days, the hybrid GLM performs a little worse than the other three models. However, the hybrid GLM is superior to the STGCN, LSTM, and RSVR models when predicting in working days, which can be seen in Figure 10b,d. The GLM model can fit the ground value much better than the other three models during the rush hours of working days (time interval 50 to 70 in Figure 10b, time interval 10 to 20, and 60 to 80 in Figure 10d, see red circle in Figure 10b,d). However, the hybrid GLM's ability to capture characters in non-rush hours of the working day is sometimes worse than that of the STGCN model (see red box in Figure 10b). We assume the reason to be the low passenger flow in the non-rush hours, which leads to less flow in the graph structures.



(a) The inflow volume in a non-working day.

(b) The inflow volume in a working day.

(c) The outflow volume in a non-working day.

(d) The outflow volume in a working day.

Figure 10. The detailed inflow and outflow volume prediction in the People's Square metro station.

To further explore the hybrid GLM's ability to predict during the rush hours of a working day, we qualified the RMSEs between the rush hours and non-rush hours in all stations. The results compared with the best baseline STGCN are shown in Table 7.

Table 7. RMSEs of the rush hours and non-rush hours for a working day in the TI 10 min.

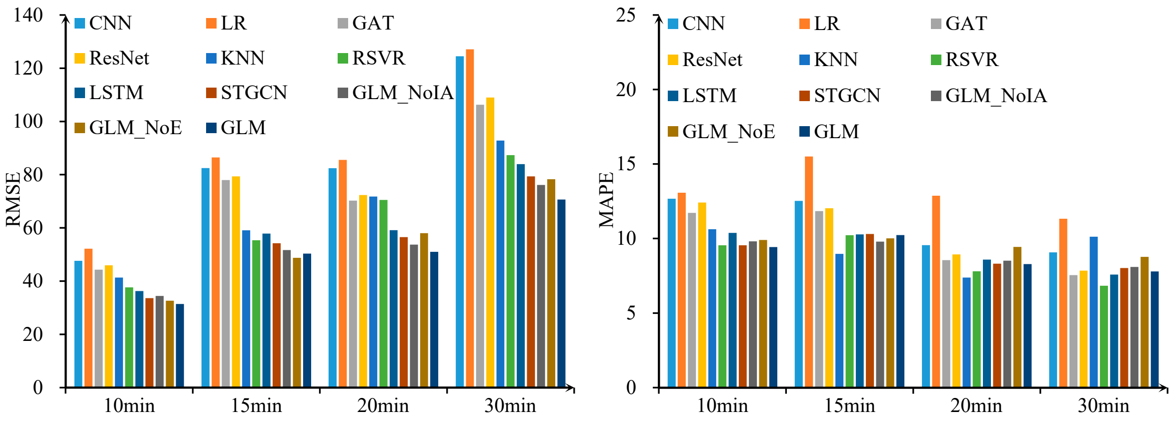
Model	Rush Hours (7:00–9:00; 11:00–13:00; 17:00–19:00)		Non-Rush Hours (6:40–7:00; 9:00–11:00; 13:00–17:00; 19:00–23:00)	
	Inflow	Outflow	Inflow	Outflow
STGCN	34.90	43.03	24.15	30.65
the hybrid GLM	31.89	38.70	23.91	32.32

As shown in Table 7, the RMSEs of the hybrid GLM are significantly smaller than STGCN in rush hours in a working day. For passenger inflow, the hybrid GLM has an 8.62% relative reduction in RMSE. For passenger outflow, the hybrid GLM brings 10.06% extra improvements compared with STGCN. While in non-rush hours, the RMSEs of the two model are relatively close. For passenger outflow, the hybrid GLM even performs worse. The results show the hybrid's ability to predict during the rush-hours of a working day is better than the best baseline STGCN. Note that the RMSEs in the rush hours are relatively higher than those in non-rush hours due to the higher passenger flow.

In summary, the hybrid GLM's prediction results during rush hours are more accurate than those of the STGCN, LSTM, and RSVR models. However, the hybrid GLM's predictive ability of non-rush hours is a little worse than the STGCN. Generally speaking, the overall predictive ability of the hybrid GLM is superior to those of STGCN, LSTM, and RSVR.

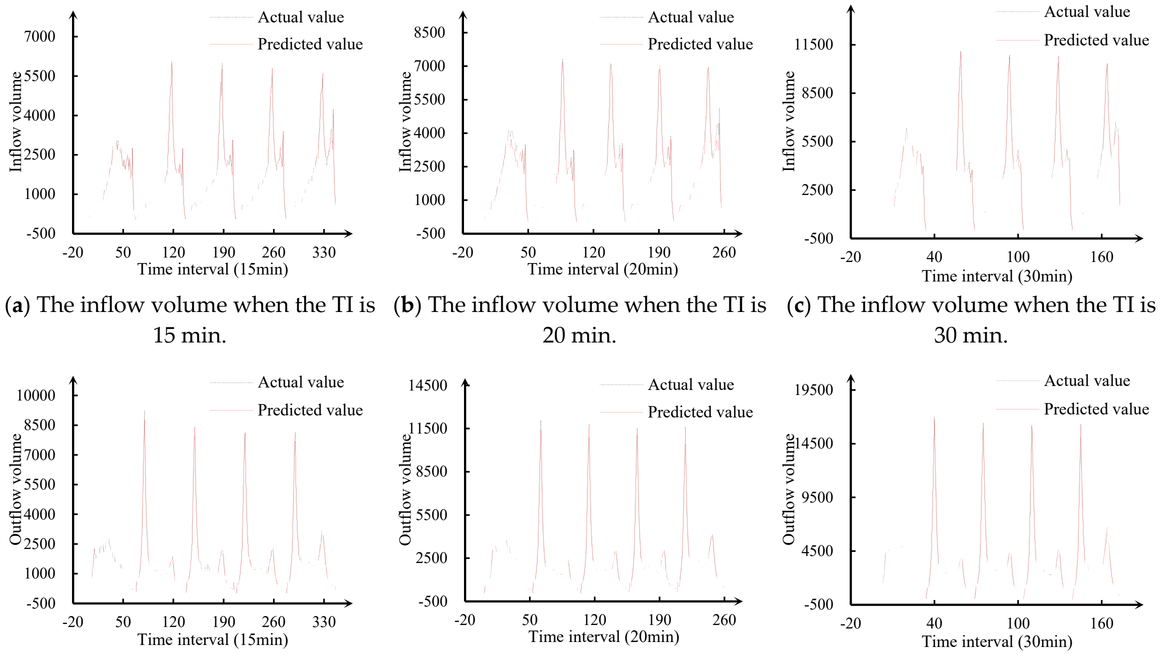
5.5.3. Prediction Performance in Different TIs

To verify the robustness of the hybrid GLM, we compared the prediction results from different time intervals (TIs). We chose 10 min, 15 min, 20 min, and 30 min as the TIs, respectively. From Figure 11 and Table 8, we can observe that the prediction precision decreases with the increasing TIs, which results from the lower number of samples in the training data. When the TI is fixed, the hybrid GLM gets the smallest RMSE compared with the other 9 baseline models. However, the MAPE of the hybrid GLM is not always the smallest. In the TI of 15 min and 20 min, KNN and RSVR get smaller MAPEs than the hybrid GLM. In the TI of 30 min, the MAPEs of RSVR and LSTM are smaller than that of the hybrid GLM. Though the MAPEs of some models are relatively small, the hybrid GLM still gets a much smaller RMSE. The reason may be that the hybrid GLM's predictive ability of rush hours is better compared with some baseline models while its predictive ability of non-rush hours is sometimes bad. As for a single TI, the conclusion made in Section 5.5.1 can be proved, too. Take TI 30 min as an example, the hybrid GLM exhibits an RMSE reduction of 43.27% and an MAPE reduction of 14.11% compared with CNN. Compared with LSTM, the hybrid GLM has a 15.87% relative reduction in RMSE. The results verify the superiority of the spatio-temporal model. The graph-structure-based model, GAT, exhibits an RMSE reduction of 2.50% and a MAPE reduction of 3.82% compared with the raster-based model, ResNet. The hybrid GLM has a 10.97% relative reduction in RMSE and a 2.99% relative reduction in MAPE compared with STGCN. The reason may be the improvement of GAT when using an asymmetric matrix. As shown in Table 8, GLM_NoIA and GLM_NoE perform a little worse than the hybrid GLM, which exhibits the effectiveness of the improved adjacent and the external factors. As shown in Figure 12, the prediction results of the hybrid GLM fit the true values well in different TIs, especially in the peak period. Note that the results of TI 10 min are shown in Figures 8 and 9. Therefore, we can conclude that the hybrid GLM outperforms the other baseline models in different TIs in total, which exhibits the robustness and high accuracy of the hybrid GLM.



(**a**) RMSE for different models and TIs. (**b**) MAPE for different models and TIs.

Figure 11. Comparison of the RMSE and MAPE for the different models and time intervals (TIs).



(**a**) The inflow volume when the TI is 15 min. (**b**) The inflow volume when the TI is 20 min. (**c**) The inflow volume when the TI is 30 min.

(**d**) The outflow volume when the TI is 15 min. (**e**) The outflow volume when the TI is 20 min. (**f**) The outflow volume when the TI is 30 min.

Figure 12. Comparison of the actual values and predicted values of the People's Square metro station in different TIs.

Table 8. Results of the RMSE and MAPE for the different models and TIs.

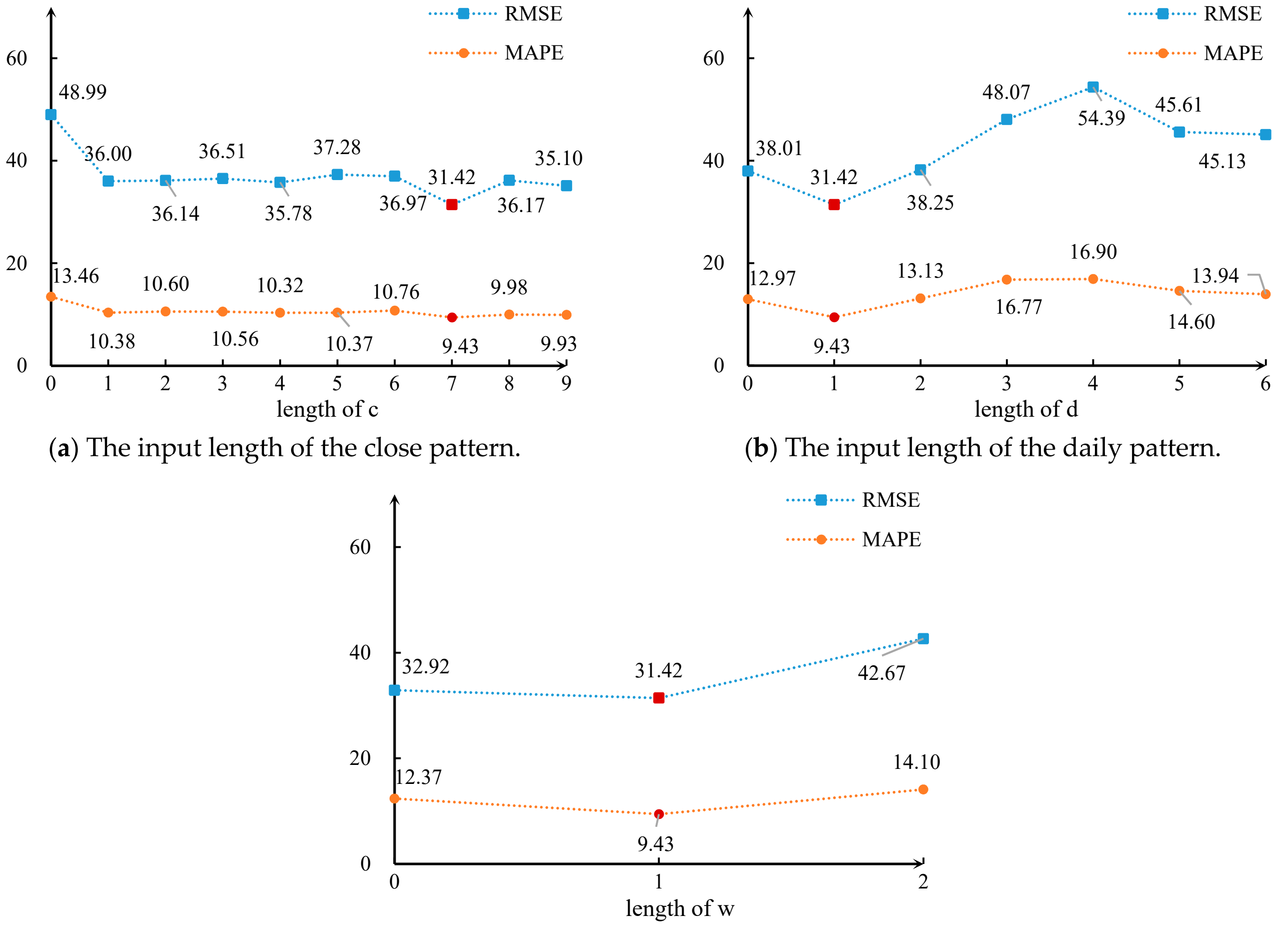
No.	TI	10 min		15 min		20 min		30 min	
	Model	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**	**RMSE**	**MAPE**
1	LR	52.18	13.08	86.46	15.51	85.54	12.87	127.09	11.33
2	CNN	47.64	12.67	82.48	12.53	82.44	9.56	124.51	9.07
3	ResNet	45.92	12.42	79.35	12.03	72.34	8.94	109.00	7.85
4	GAT	44.28	11.72	77.98	11.84	70.27	8.55	106.28	7.55
5	KNN	41.31	10.63	59.11	8.97	71.78	7.39	92.81	10.12
6	RSVR	37.67	9.55	55.33	10.22	70.49	7.81	87.35	6.83
7	LSTM	36.27	10.38	57.85	10.28	59.12	8.59	83.95	7.58
8	STGCN	33.60	10.58	54.21	10.31	56.54	8.31	79.33	8.03
9	GLM_NoIA	34.43	9.81	51.68	9.79	53.72	8.52	76.14	8.10
10	GLM_NoE	32.64	9.90	48.77	10.01	58.04	9.44	78.27	8.77
11	GLM	31.42	9.43	50.33	10.23	51.01	8.28	70.63	7.79

5.6. Parameters Tuning

The procedure for tuning the parameters is an indispensable process for the training of DL models. This section focuses on the adjustment process of some typical factors.

5.6.1. Lengths of the Different Input Patterns

We here verify the impact of the different input lengths of the three patterns, namely, the close, daily, and weekly patterns. The results are shown in Figure 13. We respectively define c, d, and w as the input length of the close, daily, and weekly patterns. Figure 13a shows the results of the effect of temporal closeness when d and w are fixed as 1 but c is changed. We can learn that the RMSE and the MAPE are very large when c is 0, which means the close pattern is very important. The best performance appears when c is 7. Figure 13b shows the results of the effect of the daily period when c is set as 7 and w is set as 1 but d varies from 0 to 6. We can observe that the RMSE and the MAPE first decrease and then increase as d increases. The optimal d is 1. Figure 13c shows the results of the effect of the weekly period when c is set as 7 and d is set as 1 but w varies from 0 to 2. From Figure 13c, we find that the RMSE and the MAPE increase when w is larger than 1, which means the situation at 7:00 a.m. the last two weeks is not closely related to that at 7:00 a.m. this week. After tuning, we can conclude that it is better to employ some temporal patterns, but the long-term trend may not be effective or even useless.



(**a**) The input length of the close pattern.

(**b**) The input length of the daily pattern.

(**c**) The input length of the weekly pattern.

Figure 13. Experimental results for the different input lengths.

5.6.2. Number of Hidden Layers

We performed nine experiments to explore the optimal depth for the hybrid GLM. We investigate the depths of the graph attention layers and the LSTM layers separately, which are denoted as D_{GAT} and D_{LSTM}. First, with D_{LSTM} = 1, D_{GAT} varies from one to three. Then the D_{LSTM} is set as 2 and 3 in the same way. The results are shown in Figure 14. Generally speaking, the RMSE and the MAPE of the model first decrease and then increase, which shows that the deeper network often has better results for capturing more characters. However, if the networks are too deep, the training may become hard, which leads to an increase in the RMSE and MAPE. More specifically, the RMSE is relatively low if the D_{LSTM} is between 1 and 2. The RMSE will increase once the D_{LSTM} becomes 3, which indicates the deeper LSTM performs worse in the model. As for D_{GAT}, the RMSE and the MAPE are relatively large when D_{GAT} is set as 1 and 3. Lastly, we find the optimal D_{LSTM} and D_{GAT} are 2 and 2.

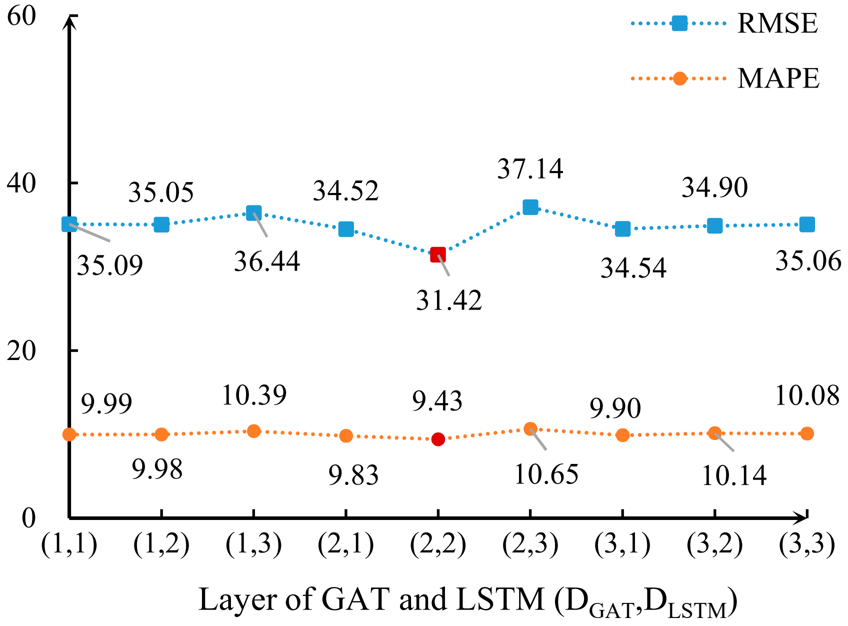
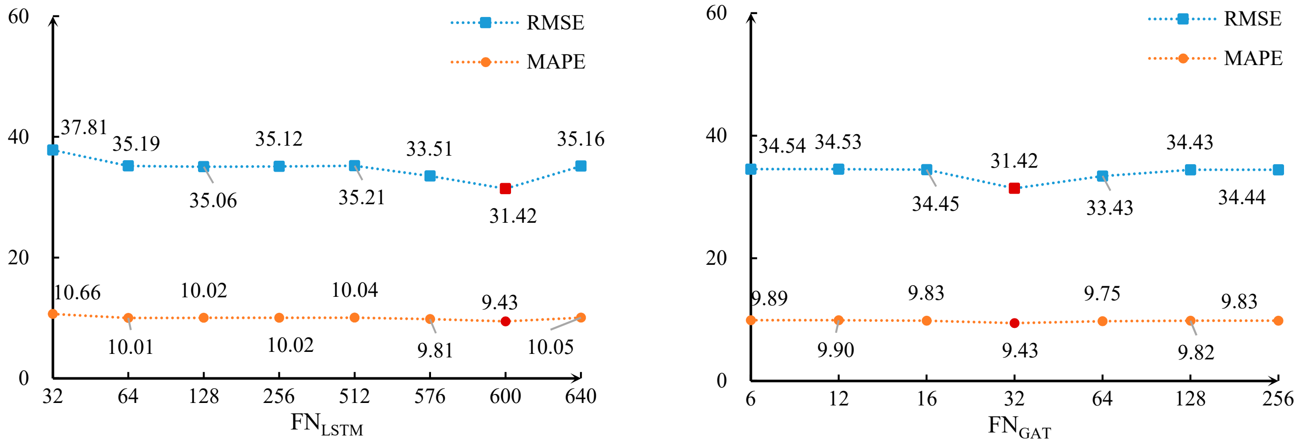


Figure 14. Experimental results for the different hidden layers.

5.6.3. Number of Hidden Neuron Units

FN_{LSTM} and FN_{GAT} represent the number of hidden units in LSTM and GAT, relatively. To test LSTM, the FN_{GAT} of the 2-layer GAT were fixed as 32 and 2, and FN_{LSTM} varies between 32, 64, 128, 256, 512,576, 600, and 640. Correspondingly, to test GAT, FN_{LSTM} was fixed as 600, and the FN_{GAT} in the first GAT layer varies between 6, 12, 16, 32, 64, 128, and 256. The results of the influence of different hidden neuron units are shown in Figure 15. Figure 15a is the result when changing FN_{LSTM}. The MAPE is relatively stable as FN_{LSTM} is changed. However, more FN_{LSTM} results in a lower RMSE up to a certain level. The turning point of FN_{LSTM} is around 600. Therefore, the optimal FN_{LSTM} is 600. Figure 15b is the result for changing FN_{GAT}. The RMSEs and the MAPEs are very stable when FN_{GAT} is changing. However, FN_{GAT} = 32 has a superior result. In summary, the model exhibits the best prediction accuracy when FN_{LSTM} is 600 and FN_{GAT} is 32.



(**a**) Different hidden layers of LSTM.

(**b**) Different hidden layers of GAT.

Figure 15. Experimental results for the different hidden layers.

6. Conclusions and Future Work

In this paper, we focus on a valuable and widely studied problem, metro passenger flow prediction, whose goal is to effectively and accurately predict the passenger flow in future time intervals for a specific region. We argue that the existing works ignore the application of asymmetric networks in a city and lose sight of the fact that further neighbors may also have some impact. In addition, some of them do not analyze the effects of external factors, such as weather, air quality, etc.

To address these issues, we propose a new method, the hybrid GLM, to predict the citywide metro passenger flow by integrating two DL methods, LSTM and GAT. By utilizing GAT, the proposed model can be used in asymmetric networks. In order to explore the influence of some nodes located a little further, we improve the adjacent matrix by applying different weights to some further neighbors. Further, LSTM structures are adopted to capture the temporal dependency and external influence, which can improve the entire model. We tested the proposed model via a case study involving the prediction of the citywide metro passenger flow in Shanghai, China for five days. The experimental results indicate that the hybrid GLM significantly outperforms several baseline models, namely LR, KNN, RSVR, LSTM, CNN, ResNet, GAT, and STGCN. A detailed comparison between the hybrid GLM and STGCN reveals that the hybrid GLM provides a higher performance of 6% to 10% for different TIs. For rush hours in a working day, the hybrid GLM fits the ground truths better, which may be more helpful for urban manager to make effective plans. The accurate prediction results can also provide references for people's traveling schedule.

However, some limitations still exist. Firstly, the prediction errors of the hybrid GLM are relatively larger than those of STGCN for the non-rush hours in a working day. Secondly, the time span of the validation datasets only covers a month, which may ignore some temporal external factors, such as seasons. Thirdly, we apply a single-step ahead prediction [58] in our study, which may cause error accumulation. In the future, we intend to address these limitations to better discover the correlations for a higher quality prediction. We will further explore external features and multi-step ahead prediction for model improvement. We also intend to investigate the application of the hybrid GLM to much longer datasets or other types of flows, such bike flow, crowd flow, and traffic flow in different TIs. Lastly, in terms of DL models, some advantages of GAT, such as capturing the bidirectional characters of the traffic lines, modeling dynamic graphs, and modeling multi-graphs, should be further studied.

Author Contributions: Conceptualization, Yong Han; Data curation, Tongxin Peng; Formal analysis, Tongxin Peng and Cheng Wang; Funding acquisition, Yong Han; Investigation, Tongxin Peng; Methodology, Tongxin Peng and Cheng Wang; Project administration, Yong Han; Resources, Yong Han; Software, Tongxin Peng; Supervision, Ge Chen; Validation, Tongxin Peng; Visualization, Tongxin Peng and Zhihao Zhang; Writing—original draft, Tongxin Peng; Writing—review & editing, Tongxin Peng and Cheng Wang. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Natural Science Foundation of Shandong Province, China (Grant No. ZR2020MD020).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable to this article.

Acknowledgments: Thanks for the data provided by Shanghai Public Transportation Group.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Zhu, D.; Huang, Z.; Shi, L.; Wu, L.; Liu, Y. Inferring spatial interaction patterns from sequential snapshots of spatial distributions. *Int. J. Geogr. Inf. Sci.* **2018**, *32*, 783–805. [CrossRef]
2. Wei, Y.; Chen, M.C. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transp. Res. Part C Emerg. Technol.* **2012**, *21*, 148–162. [CrossRef]
3. Dhall, D.; Kaur, R.; Juneja, M. Machine Learning: A Review of the Algorithms and Its Applications. In Proceedings of the ICRIC 2019, Cham, Switzerland, 2019; pp. 47–63.
4. Sha, S.; Li, J.; Zhang, K.; Yang, Z.; Wei, Z.; Li, X.; Zhu, X. RNN-Based Subway Passenger Flow Rolling Prediction. *IEEE Access* **2020**, *8*, 15232–15240. [CrossRef]
5. Yang, D.; Yang, H.; Wang, P.; Li, S. Multi-Step Traffic Flow Prediction Using Recurrent Neural Network. In Proceedings of the 2019 IEEE International Conferences on Ubiquitous Computing & Communications (IUCC) and Data Science and Computational Intelligence (DSCI) and Smart Computing, Networking and Services (SmartCNS), Shenyang, China, 21–23 October 2019; pp. 803–808.
6. Qiao, Y.; Wang, Y.; Ma, C.; Yang, J. Short-term traffic flow prediction based on 1DCNN-LSTM neural network structure. *Mod. Phys. Lett. B* **2020**, *35*, 2150042. [CrossRef]
7. Ren, Y.B.; Chen, H.F.; Han, Y.; Cheng, T.; Zhang, Y.; Chen, G. A hybrid integrated deep learning model for the prediction of citywide spatio-temporal flow volumes. *Int. J. Geogr. Inf. Sci.* **2020**, *34*, 802–823. [CrossRef]
8. Guo, S.; Lin, Y.; Feng, N.; Song, C.; Wan, H. Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 922–929. [CrossRef]
9. Lv, M.; Hong, Z.; Chen, L.; Chen, T.; Zhu, T.; Ji, S. Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction. *IEEE Trans. Intell. Transp.* **2020**, 1–12. [CrossRef]
10. Ye, J.; Zhao, J.; Ye, K.; Xu, C. Multi-STGCnet: A Graph Convolution Based Spatial-Temporal Framework for Subway Passenger Flow Forecasting. In Proceedings of the 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 19–24 July 2020; pp. 1–8.
11. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the 30th International Conference on Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
12. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. *arXiv* **2017**, arXiv:1710.10903.
13. Liu, S.Y.; Liu, S.; Tian, Y.; Sun, Q.L.; Tang, Y.Y. Research on Forecast of Rail Traffic Flow Based on ARIMA Model. *J. Phys. Conf. Ser.* **2021**, *1792*, 012065. [CrossRef]
14. Cheng, T.; Wang, J.; Haworth, J.; Heydecker, B.; Chow, A. A Dynamic Spatial Weight Matrix and Localized Space—Time Autoregressive Integrated Moving Average for Network Modeling. *Geogr. Anal.* **2014**, *46*, 75–97. [CrossRef]
15. Milenković, M.; Švadlenka, L.; Melichar, V.; Bojović, N.; Avramović, Z. SARIMA modelling approach for railway passenger flow forecasting. *Transp. Vilnius* **2018**, *33*, 1113–1120. [CrossRef]
16. Smith, B.L.; Williams, B.M.; Keith Oswald, R. Comparison of parametric and nonparametric models for traffic flow forecasting. *Transp. Res. Part C Emerg. Technol.* **2002**, *10*, 303–321. [CrossRef]
17. Jiao, P.; Li, R.; Sun, T.; Hou, Z.; Ibrahim, A. Three Revised Kalman Filtering Models for Short-Term Rail Transit Passenger Flow Prediction. *Math. Probl. Eng.* **2016**, *2016*, 9717582. [CrossRef]
18. Ding, C.; Duan, J.; Zhang, Y.; Wu, X.; Yu, G. Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility. *IEEE Trans. Intell. Transp.* **2018**, *19*, 1054–1064. [CrossRef]

19. Liang, S.; Ma, M.; He, S.; Zhang, H. Short-Term Passenger Flow Prediction in Urban Public Transport: Kalman Filtering Combined K-Nearest Neighbor Approach. *IEEE Access* **2019**, *7*, 120937–120949. [CrossRef]
20. Suthaharan, S. Support Vector Machine. In *Machine Learning Models and Algorithms for Big Data Classification: Thinking with Examples for Effective Learning*; Suthaharan, S., Ed.; Springer: Boston, MA, USA, 2016; pp. 207–235. [CrossRef]
21. Wang, X.; Zhang, N.; Zhang, Y.; Shi, Z. Forecasting of Short-Term Metro Ridership with Support Vector Machine Online Model. *J. Adv. Transp.* **2018**, *2018*, 3189238. [CrossRef]
22. Zhang, Y.; Liu, Y. Traffic forecasting using least squares support vector machines. *Transportmetrica* **2009**, *5*, 193–213. [CrossRef]
23. Sun, Y.X.; Leng, B.; Guan, W. A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system. *Neurocomputing* **2015**, *166*, 109–121. [CrossRef]
24. Chen, Q.A.; Li, W.Q.; Zhao, J.H. The Use of Ls-Svm for Short-Term Passenger Flow Prediction. *Transp. Vilnius* **2011**, *26*, 5–10. [CrossRef]
25. Li, W.; Sui, L.; Zhou, M.; Dong, H. Short-term passenger flow forecast for urban rail transit based on multi-source data. *Eurasip. J. Wirel. Commun. Netw.* **2021**, *2021*, 9. [CrossRef]
26. Cao, C.; Xu, J. Short-Term Traffic Flow Predication Based on PSO-SVM. In Proceedings of the First International Conference on Transportation Engineering, Chengdu, China, 22–24 July 2007; pp. 167–172.
27. Roos, J.; Bonnevay, S.; Gavin, G. Short-Term Urban Rail Passenger Flow Forecasting: A Dynamic Bayesian Network Approach. In Proceedings of the 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), Anaheim, CA, USA, 18–20 December 2016; pp. 1034–1039.
28. Liu, L.; Chen, R.-C.; Zhao, Q.; Zhu, S. Applying a multistage of input feature combination to random forest for improving MRT passenger flow prediction. *J. Ambient. Intell. Humaniz. Comput.* **2019**, *10*, 4515–4532. [CrossRef]
29. Zhang, S.; Liu, Z.; Shen, F.; Wang, S.; Yang, X. A prediction model of buses passenger flow based on neural networks. *J. Phys. Conf. Ser.* **2020**, *1656*, 012002. [CrossRef]
30. Bai, J.; He, M.; Shuai, C. Short-Term Passenger Flow Forecast in Urban Rail Transit Based on Enhanced K-Nearest Neighbor Approach. *CICTP* **2019**, *2019*, 1695–1706.
31. Paul, S.; Singh, L. A review on advances in deep learning. In Proceedings of the 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI), Kanpur, India, 14–17 December 2015; pp. 1–6.
32. Connor, J.T.; Martin, R.D.; Atlas, L.E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **1994**, *5*, 240–254. [CrossRef]
33. Zhang, J.L.; Chen, F.; Shen, Q. Cluster-Based LSTM Network for Short-Term Passenger Flow Forecasting in Urban Rail Transit. *IEEE Access* **2019**, *7*, 147653–147671. [CrossRef]
34. Cho, K.; Merrienboer, B.V.; Gulcehre, C.; Ba Hdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *arXiv* **2014**, arXiv:1406.1078.
35. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
36. Zhang, D.; Kabuka, M.R. Combining Weather Condition Data to Predict Traffic Flow: A GRU Based Deep Learning Approach. *IET Intell. Transp. Syst.* **2018**, *12*, 578–585. [CrossRef]
37. Han, Y.; Wang, C.; Ren, Y.; Wang, S.; Zheng, H.; Chen, G. Short-Term Prediction of Bus Passenger Flow Based on a Hybrid Optimized LSTM Network. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 366. [CrossRef]
38. Lin, S.; Tian, H. Short-Term Metro Passenger Flow Prediction Based on Random Forest and LSTM. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; pp. 2520–2526.
39. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. *Commun. ACM* **2012**, *25*, 1097–1105.
40. Zhang, J.; Zheng, Y.; Qi, D.; Li, R.; Yi, X. DNN-based prediction model for spatio-temporal data. In Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Burlingame, CA, USA, 31 October–3 November 2016; p. 92.
41. Zhang, J.B.; Zheng, Y.; Qi, D.K.; Li, R.Y.; Yi, X.W.; Li, T.R. Predicting citywide crowd flows using deep spatio-temporal residual networks. *Artif. Intell.* **2018**, *259*, 147–166. [CrossRef]
42. Yu, F.; Wei, D.; Zhang, S.; Shao, Y. 3D CNN-based Accurate Prediction for Large-scale Traffic Flow. In Proceedings of the 2019 4th International Conference on Intelligent Transportation Engineering (ICITE), Singapore, 5–7 September 2019; pp. 99–103.
43. Bruna, J.; Zaremba, W.; Szlam, A.; Lecun, Y. Spectral Networks and Locally Connected Networks on Graphs. *arXiv* **2013**, arXiv:1312.6203.
44. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
45. Yu, B.; Yin, H.; Zhu, Z. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv* **2017**, arXiv:1709.04875.
46. Zhao, L.; Song, Y.; Zhang, C.; Liu, Y.; Wang, P.; Lin, T.; Deng, M.; Li, H. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Trans. Intell. Transp.* **2020**, *21*, 3848–3858. [CrossRef]
47. Zhang, J.L.; Chen, F.; Guo, Y.A.; Li, X.H. Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. *IET Intell. Transp. Syst.* **2020**, *14*, 1210–1217. [CrossRef]

48. Zhang, J.; Chen, F.; Cui, Z.; Guo, Y.; Zhu, Y. Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. *IEEE Trans. Intell. Transp.* **2020**, 1–11. [CrossRef]
49. Huang, Y.; Bi, H.; Li, Z.; Mao, T.; Wang, Z. STGAT: Modeling Spatial-Temporal Interactions for Human Trajectory Prediction. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 6271–6280.
50. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
51. Daldal, N.; Sengur, A.; Polat, K.; Cömert, Z. A novel demodulation system for base band digital modulation signals based on the deep long short-term memory model. *Appl. Acoust.* **2020**, *166*, 107346. [CrossRef]
52. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press: Cambridge, UK, 2016; Volume 1.
53. Ober, P.B. Introduction to linear regression analysis. *J. R. Stat. Soc.* **2010**, *40*, 2775–2776. [CrossRef]
54. Xu, Y.; Li, D. Incorporating Graph Attention and Recurrent Architectures for City-Wide Taxi Demand Prediction. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 414. [CrossRef]
55. Zhang, L.; Liu, Q.; Yang, W.; Wei, N.; Dong, D. An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction. *Procedia Soc. Behav. Sci.* **2013**, *96*, 653–662. [CrossRef]
56. Ma, X.; Dai, Z.; He, Z.; Ma, J.; Wang, Y.; Wang, Y. Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction. *Sensors* **2017**, *17*, 818. [CrossRef] [PubMed]
57. Han, Y.; Wang, S.K.; Ren, Y.B.; Wang, C.; Gao, P.; Chen, G. Predicting Station-Level Short-Term Passenger Flow in a Citywide Metro Network Using Spatiotemporal Graph Convolutional Neural Networks. *ISPRS Int. J. Geo-Inf.* **2019**, *8*, 243. [CrossRef]
58. Landassuri-Moreno, V.M.; Bustillo-Hernández, C.L.; Carbajal-Hernández, J.J.; Fernández, L.P.S. Single-Step-Ahead and Multi-Step-Ahead Prediction with Evolutionary Artificial Neural Networks. In Proceedings of the Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications, Havana, Cuba, 20–23 November 2013; pp. 65–72.