



Short-term origin–destination flow prediction for urban rail network: a deep learning method based on multi-source big data

Hongmeng Cui¹ · Bingfeng Si¹ · Jiayuan Wang¹ · Ben Zhao² · Weiting Pan³

Received: 13 September 2023 / Accepted: 12 February 2024
© The Author(s) 2024

Abstract

Short-term prediction of origin–destination (OD) flow is a primary but complex assignment to urban rail companies, which is the basis of intelligent and real-time urban rail transit (URT) operation and management. The short-term prediction of URT OD flow has three special characteristics: data lag, data dimensionality, and data malconformation, distinguishing it from other short-term prediction tasks. It is essential to propose a novel prediction algorithm that considers the special characteristics of the URT OD flow. For this purpose, based on deep learning methods and multi-source big data, a modified spatial–temporal long short-term memory (ST-LSTM) model is established. The proposed model comprises four components: (1) a temporal feature extraction module is devised to extract time information within network-wide historical OD data; (2) a spatial correlation learning module is introduced to address the data malconformation and data dimensionality problems, which provides an interpretable spatial correlation quantization method; (3) an input control-gated mechanism is originally proposed to solve the data lag problem, which combines the processed available OD flow and real-time inflow/outflow; (4) a fusion module combines historical spatial–temporal features with real-time information to achieve accurate OD flow prediction. We also further discuss the interpretability of the model in detail. The ST-LSTM model is evaluated by sufficient experiments on two large-scale actual subway datasets from Nanjing and Beijing, and the experimental results demonstrate that it can better learn the spatial–temporal correlations and exceed the rest benchmarking methods.

Keywords Short-term origin–destination prediction · Deep learning · Urban rail transit · Spatial–temporal correlations

Introduction

With the urban rail transit (URT) system becoming increasingly complex, its intelligent construction has attracted substantial attention. As one of the essential tasks of intelligent transportation systems (ITS), short-term origin–destination (OD) forecasting has generated more and more research interest due to its realistic impact on both operators and passengers [53]. For metro operators, accurate OD prediction results are helpful to better monitor the dynamic spatial–temporal distribution of passengers, hence providing support for the decisions on network management tasks. Accurate prediction results of short-term OD flow can also contribute to the appropriate staffing in stations, according to which the operators can reasonably evacuate passengers in advance to avoid or improve accidents that can damage numerous passengers. Moreover, for passengers, good prediction results of short-term OD flow are conducive to arranging travel routes rationally, thus saving travel time and improving the travel experience.

✉ Bingfeng Si
bfsi@bjtu.edu.cn

Hongmeng Cui
13654081957@163.com

Jiayuan Wang
15910531682@163.com

Ben Zhao
1012379660@qq.com

Weiting Pan
13019606829@163.com

- ¹ School of Systems Science, Beijing Jiaotong University, Beijing, China
- ² Shanxi Intelligent Transportation Institute Co., Ltd, Taiyuan, Shanxi, China
- ³ School of Traffic and Transportation, Beijing Jiaotong University, Beijing, China

The short-term passenger flow prediction of URT at the network-scale level can be categorized into inflow/outflow prediction, OD flow prediction, and passenger flow distribution prediction [30, 55]. Inflow/outflow prediction refers to predicting the passenger flow entering or leaving stations, which has been extensively studied [21, 44, 45]. Short-term OD prediction can be performed after obtaining the real-time inflow/outflow passenger flow data [53]. Finally, passenger flow distribution prediction aims to predict the specific travel path passengers select from the starting station to the terminal station. Since acquiring individual travel trajectories in URT is challenging, the OD flow is commonly used as the fundamental input, and the traffic assignment model is employed to estimate travelers' route choice behavior. In summary, OD flow prediction acts as a bridge between inflow/outflow prediction and passenger flow distribution prediction, playing a crucial role in short-term URT prediction at the network-scale level. Accurate OD flow prediction results can provide valuable spatial-temporal flow patterns between subway stations and contribute to a better understanding of passenger travel behavior [40]. Therefore, this paper studies the short-term URT OD flow prediction.

Heretofore, short-term traffic forecasting has been studied extensively. Various prediction models and algorithms have been proposed to predict road traffic flow, ride-hailing demand, and the inflow/outflow of URT. As far as we know, there is limited research on short-term URT OD flow, primarily due to some characteristics that set URT OD flow forecasting tasks apart from other tasks. The unique characteristics of URT OD flow pose significant challenges in current research. These characteristics are listed as follows:

- **Data lag.** Real-time traffic data are accessible for the majority of short-term traffic prediction tasks. For example, in the ride-hailing OD demand forecasting task, the real-time OD demand can be obtained as users are required to specify the origin and destination when they initiate the services [13]. Likewise, we can also obtain the URT inflow/outflow and the traffic speed in real-time. Significantly, when predicting real-time road OD demand, the prediction model typically relies on road traffic volumes as input, which can be acquired in real time [40]. For these tasks, when the short-term traffic prediction is conducted, the practical traffic data within the past several time intervals are available. Nevertheless, since it always takes travel time from the origin to the destination, the real-time URT OD data are unavailable. The OD matrix is available only after all passengers arrive at the destination and swipe their cards to exit the station.
- **Data dimensionality.** Assuming that n represents the total number of stations within the URT system, and the number of OD flows is n^2 . This dramatically increases

the difficulty of forecasting OD flows. For instance, in the case of Beijing, there are 405 URT stations and the dimensionality of the OD flow to a staggering 164025. Thus, it is evident that the dimensionality of the OD flow far surpasses the cardinality of transportation networks.

- **Data malconformation.** Because OD flows have large dimensionality, the total traffic gets distributed across various OD pairs, resulting in an extremely imbalanced passenger flow distribution, and many OD pairs have either small or zero flows. Although road traffic and ride-hailing services' OD predictions also face data malformation issues, they are less severe than those encountered in rail transit. This is because of the relatively large traffic volume and the availability of real-time passenger flow information. On the other hand, there is a significant volume of traffic in the corresponding URT inflow/outflow, road flow, and ride-hailing demand. These areas experience dense traffic and have relatively high passenger volumes.

Table 1 presents the features of various traffic flow prediction techniques. It is evident that the URT OD flow prediction task stands out as it necessitates meticulous model input design while also facing challenges related to data dimensionality and malformation. Despite the valuable insights offered by URT OD demand, research on short-term OD flow prediction specifically for URT is scarce.

In summary, this paper is driven by the need to tackle challenges encountered in short-term OD prediction for URT.

- First, the travel patterns are time-varying. For instance, there are notable distinctions in travel patterns between weekdays and weekends. Thus, the time information should be explicitly considered when predicting real-time OD flow.
- Second, obtaining the real-time OD matrix is not feasible. Using OD matrices from the last few time spans as model inputs is impractical. Therefore, determining the appropriate range of input historical data becomes the second challenge in real-time OD prediction.
- Third, existing methods often overlook the correlation between inflows/outflows and OD flows in URT. Therefore, it is necessary to explicitly consider their relationships when developing forecasting models.
- Fourth, most previous deep learning studies in OD flow prediction learn spatial correlations through CNN-based methods. These methods consider one OD flow to be influenced by all the other OD flows [3, 36, 53] or by its local neighboring OD flows [13, 46], which significantly reduces the prediction accuracy by introducing excessively irrelevant information or failing to extract the global spatial correlations. Therefore, how to accurately extract spatial correlations while considering data dimen-

Table 1 Summary of the features of various traffic flow prediction techniques

Prediction method	Data lag	Data dimensionality	Data malconformation	References
Road flow	✓	✓	✓	Wang et al. [35] and Guo et al. [5]
Road speed	✓	✓	✓	Ma et al. [24, 25]
Road OD demand	✓	•	•	Lu et al. [22]
Ride-hailing OD demand	✓	•	•	Chu et al. [3]
Ride-hailing demand	✓	✓	✓	Jin et al. [11] and Wang et al. [33]
URT inflow/outflow	✓	✓	✓	Han et al. [6] and Zhang et al. [54]
URT OD flow	•	•	•	This study

“✓” indicates issue not present, while “•” represents issue present

sionality and malconformation issues is another crucial issue.

- Finally, in the existing literature, the time interval for short-term OD flow prediction ranges from 5 min to 1 week. The choice of prediction time span directly impacts the accuracy of the forecasts. Consequently, determining the appropriate prediction time interval is another crucial aspect that requires further exploration.

To tackle these challenges, this paper suggests a solution in the form of an enhanced long short-term memory (LSTM) model called spatial–temporal long short-term memory (ST-LSTM). The model learns spatial–temporal correlations from multi-source data across the network to provide short-term OD flow predictions. Here, ‘spatial correlation’ means the internal relationship and mutual influence between different origin–destination pairs, and ‘temporal correlation’ refers to the changing relationships of OD flows across different time periods and the interactions and influences between those periods. In the proposed model, we establish a temporal feature extraction module to capture all the necessary time information for subsequent processes. We originally introduce a data inflow control gate to determine the scope of historical data and available OD data and enable the integration of OD flow information with real-time inflow/outflow data, effectively addressing the issue of data lag. Considering the data dimensionality and malconformation issues, we innovatively introduce an interpretable spatial correlations learning method, which leverages passenger flow data, OD matrices, and URT network geographic information to extract spatial correlations among OD pairs in URT systems. A comprehensive index is established to assess the spatial correlations between OD pairs across the entire network, considering the impact of trend changes, passenger flow interactions, and distance distribution. Subsequently, based on the index parameters, OD pairs are ranked in descending order. During the prediction process, the data of ODs with a strong correlation is selected as part of the input. By evaluating the spatial correlation among ODs within the entire network, we effectively address the issue of incomplete infor-

mation extraction. Additionally, in the prediction phase, we take into account pre-selected strongly correlated OD pairs to tackle the problem of introducing redundant data. Furthermore, this paper examines prediction time intervals of 15 min, 30 min, and 60 min, discussing the forecast results for each time scale. Experiments conducted on real-world datasets from the Nanjing Subway and Beijing Subway demonstrate the effectiveness of the ST-LSTM model. In summary, the main contributions of this work can be summarized as follows:

- We provide an overview of the characteristics and challenges associated with short-term OD prediction in URT. Additionally, we introduce a deep learning-based OD flow prediction model that explicitly takes into account the unique attributes of URT.
- We propose a temporal feature extraction module to extract time information and select historical data for training.
- We introduce an input control-gated mechanism that helps determine the available real-time OD data range. This mechanism also enables the aggregation of OD flow information and real-time inflow/outflow information.
- In the spatial correlation learning module, we not only account for the spatial relationships within a fixed topology, but also consider the characteristics of OD flow to extract dynamic spatial correlations across the entire network.
- We perform extensive experiments on two real-world subway datasets to identify the suitable prediction time interval and showcase the effectiveness of our approach.

The rest of this paper is structured as follows. In “[Related work](#)” section, we provide a review of the relevant literature. Next, in “[Preliminaries](#)” section, we define key terms and explain LSTM. The methodology utilized in this study is outlined in “[Methodologies](#)” section, followed by the presentation of experimental details and results in “[Experiment](#)” section. Finally, the conclusion of our work is presented in the last section.

Related work

Traffic inflow and outflow predictions

The prediction of traffic inflow and outflow has received extensive attention in research. Various models have been proposed, ranging from traditional statistical models to artificial intelligence-based models. The latter category has shown greater effectiveness in practical applications due to the availability of large-scale passenger travel data and advancements in deep learning techniques. Since Ma et al. [24] first applied LSTM networks to the field of traffic prediction in 2015, numerous deep learning models have been introduced, including Stacked Auto-Encoder (SAE) [23], classical CNN-based models [25], and ST-GCN [49]. It is hard to say that one approach is consistently superior to the others in any situation. Therefore, researchers are increasingly exploring combinations of two or more models, such as ConvLSTM [21], SA-DNN [32], and 1DCNN-LSTM-Attention [35]. Some of these models are designed for individual or multiple subway stations [19, 45], while others aim at network-wide predictions [49]. Some utilize historical static data for prediction [2], while others leverage real-time dynamic data [47].

Overall, different models are developed to cater to various scenarios. However, all of these models focus on traffic inflow or outflow forecasting and differ significantly from OD flow prediction in terms of data lag, data dimensionality, and data malformation, as mentioned in the introduction section. Therefore, it is essential to establish prediction models that account for URT OD flows' unique characteristics.

Traffic OD flow prediction

Obtaining accurate OD flow predictions is significantly more challenging than predicting traffic inflow or outflow due to data dimensionality and malformation issues. This applies to various OD flows, such as road OD flow, ride-hailing OD flow, and URT OD flow.

As mentioned in the introduction section, OD flow prediction differs from traffic inflow or outflow predictions. Based on the prediction models and prediction objectives, related studies can be categorized as follows.

Regarding the prediction models, OD flow prediction can be classified into traditional methods and deep learning methods. A traditional method involves treating OD flow as time series data [34] and applying commonly used time-series algorithms. Li et al. [17] developed an enhanced ARIMI-based model to predict passenger demand variations within hotspot areas. Moreira-Matias et al. [28] integrated ARIMA model, weighted time-varying Poisson model and time-varying Poisson model, which belong to time-series analysis methods, to forecast passenger demand. Other traditional methods utilize classical machine learning

approaches such as back-propagation neural networks and support vector machine (SVM). Li et al. [18] established the least squares support vector machine (LS-SVM) to predict short-term traffic demand. However, these traditional methods fail to capture nonlinear spatial-temporal relationships accurately, leading to significant prediction errors. Additionally, they consume substantial computing resources and pose challenges for practical applications.

To address these issues, deep learning methods have been successfully applied to various research domains in recent years. Some researchers employed the LSTM model to forecast OD flow. Based on the LSTM network, Ye et al. [48] predicted the OD flow of various transportation modes using multi-source data. Jiang et al. [10] combined LSTM with the recursive Bayesian to capture the dynamic temporal correlations of OD flows. However, these approaches do not consider the differentiation of passenger travel patterns in historical data, introducing noise while learning temporal correlations and thus reducing the model's performance. To address this issue, Yang et al. [42] constructed an improved LSTM model (ELF-LSTM) to enhance the ability to capture long-term temporal features, focusing only on the temporal relationships between OD flows in the same time period (daily/weekly intervals). However, this method is only effective when a sufficiently large amount of data are available. Additionally, the above methods do not explicitly consider spatial relationships. To simultaneously capture the spatial and temporal correlations among all OD pairs, Ke et al. [12] utilized a ConvLSTM model. They stacked multiple ConvLSTM layers and CNN layers to handle historical OD demand data and travel time data, respectively. Similarly, Chu et al. [3] developed a multi-scale convolutional LSTM (MultiConvLSTM) network to predict taxi demand. Bai et al. [1] employed a cascade graph convolutional recurrent neural network to capture spatial-temporal relationships within passenger demand data across the entire city. However, these methods have the disadvantage of including redundant data from weakly correlated regions. This introduces irrelevant information during the extraction of spatial correlations, leading to decreased forecasting accuracy. To address this issue, Yao et al. [46] developed a "local CNN" model that extracts spatial relationships only within geographically close regions. Ke et al. [13] proposed a spatial-temporal model for predicting ride-sourcing OD demand, which captures spatial correlations only within OD pairs with close origins and/or destinations. However, these methods overlook the hidden relationships between geographically distant regions.

Concerning the prediction objectives, short-term OD flow predictions can be categorized into road OD flow prediction [1], taxi OD flow prediction [3, 33], bus OD flow prediction [50, 51], and URT OD flow prediction [15, 31, 39, 42]. Different prediction objectives result in variations in data

availability. In road networks, obtaining real-time OD flow data or actual OD flow data is impossible. However, real-time cross-section passenger flow is available, and an optimization model can be used to estimate the road OD flow. Due to the lack of real OD flow data for comparison, evaluating the accuracy of predicted road OD flow becomes challenging [43]. For taxi OD demand, existing methods typically divide the entire research area into transportation analysis zones (TAZs) due to the absence of fixed pick-up and drop-off points. The real OD flow between TAZs can be obtained in such cases, but real-time OD flow data is unavailable. In the bus system, due to the extensive scale of bus networks, existing research usually focuses on predicting the OD flow for one or several routes. Moreover, data availability differs among bus systems, as some systems record boarding and alighting stations, while others only record boarding stations. In the URT network, the real OD flow can be obtained based on the swiping records of smart cards, as the subway stations are fixed and passengers must swipe their cards when entering and exiting the station. However, as mentioned in the previous section, real-time OD flow is unavailable due to the time passengers travel from the inbound to the outbound station.

In summary, some of the existing models overlook the temporal variability of passenger travel patterns, and some have issues with redundant or insufficient information extraction in capturing spatial correlations. In URT, research on using deep learning methods to predict OD flow is limited, whereas most models are more suitable for road networks. Therefore, developing an OD flow prediction model specifically tailored to URT systems that can effectively extract spatial-temporal correlations is necessary.

Preliminaries

The objective of this study is to predict the OD flow in the next time slot using historical information, with a time span defined as 15 min in this work. The OD flow M and inflow/outflow F can be obtained from automatic fare collection (AFC) data in URT. Assuming an urban rail transit network consists of n stations and a day is divided into H time slots. Our prediction target is the OD volume in the time slot t on day d . We can represent M and F as one-dimensional time series, as shown in Eqs. (1)–(5). It is worth noting that the OD flow in each time slot is based on the time slot in which passengers enter the stations, as passengers entering at the same time may have different exit times. The inflow/outflow sequences are extracted based on the corresponding passenger entry and exit times.

$$M_{ij}^d = \{m_{ij}^{d,t-h}, m_{ij}^{d,t-h+1}, \dots, m_{ij}^{d,t-1}, m_{ij}^{d,t}\}, \quad (1)$$

$$M_{ij}^{d-y} = \{m_{ij}^{d-y,1}, m_{ij}^{d-y,2}, \dots, m_{ij}^{d-y,H-1}, m_{ij}^{d-y,H}\}, \quad (2)$$

$$, y = 1, 2, \dots, d - 1,$$

$$F_i^d = \{f_i^{d,t-h}, f_i^{d,t-h+1}, \dots, f_i^{d,t-1}, f_i^{d,t}\}, \quad (3)$$

$$F_i^{d-y} = \{f_i^{d-y,1}, f_i^{d-y,2}, \dots, f_i^{d-y,H-1}, f_i^{d-y,H}\}, \quad (4)$$

$$, y = 1, 2, \dots, d - 1,$$

$$f_i^{d,t} = \sum_j^n m_{ij}^{d,t} (\text{inflow}) \quad \text{or} \quad f_j^{d,t} = \sum_i^n m_{ij}^{d,t} (\text{outflow}), \quad (5)$$

where $m_{ij}^{d,t}$ denotes the OD flow from station i to station j in the time slot t on day d . $\sum_j^n m_{ij}^{d,t}$ is the sum of OD flows entering station i in the time slot t on day d , and $f_i^{d,t}$ represents the inflow/outflow entering/exiting station i in the time slot t on day d . It is important to note that there are strong correlations between OD flows and inflows/outflows, as shown in Eq. (5). The inflow/outflow for a station equals the sum of all corresponding inbound/outbound OD flows from that station.

Passengers' travel patterns exhibit variations based on both time and location. Therefore, to achieve highly accurate prediction results, it is necessary to consider not only the temporal characteristics of OD flow but also capture the spatial correlation between OD pairs. In the URT network, the passenger flow of different OD pairs can influence each other. For instance, a positive correlation may exist between passenger flows from the same station to different stations since they are all influenced by outflows from the same station. Furthermore, when the originating and terminating stations are situated in areas with similar functions (e.g., residential or commercial areas), passenger flows between different originating and terminating stations may still demonstrate similar travel patterns. Taking into account these considerations, we design a matrix that incorporates both the temporal and spatial features of the historical data for different OD flows. In Eq. (6), each row of the matrix represents the temporal aspect of one OD flow, while each column represents the spatial aspect.

$$M = \begin{bmatrix} M_{11}^1 & M_{12}^1 & \dots & M_{11}^{d-y} & \dots & M_{11}^{d-1} & M_{11}^d \\ M_{12}^1 & M_{12}^2 & \dots & M_{12}^{d-y} & \dots & M_{12}^{d-1} & M_{12}^d \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ M_{nn}^1 & M_{nn}^2 & \dots & M_{nn}^{d-y} & \dots & M_{nn}^{d-1} & M_{nn}^d \end{bmatrix}_{n^2 \times d} \quad (6)$$

$$= \begin{bmatrix} m_{11}^{1,1} & m_{11}^{1,2} & \dots & m_{11}^{d-y,H-1} & m_{11}^{d-y,H} & \dots & m_{11}^{d,t} \\ m_{12}^{1,1} & m_{12}^{1,2} & \dots & m_{12}^{d-y,H-1} & m_{12}^{d-y,H} & \dots & m_{12}^{d,t} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ m_{nn}^{1,1} & m_{nn}^{1,2} & \dots & m_{nn}^{d-y,H-1} & m_{nn}^{d-y,H} & \dots & m_{nn}^{d,t} \end{bmatrix}_{n^2 \times [H(d-1)+1]}$$

Furthermore, predicting one flow using all the historical passenger data from every OD pair in the URT network is

unnecessary and overly complicated. For instance, in the case of the Beijing metro, travel patterns differ significantly between weekdays and weekends, and festivals also significantly impact traveler behavior. Additionally, some OD flows exhibit little correlation due to their corresponding stations belonging to different railway lines and having considerable physical distances between them. Therefore, it is crucial to select historical data that has similar travel patterns to the predicted day and identify in advance the OD pairs that correlate with the OD pair for which the passenger flow volume is being predicted. This approach allows us to build a matrix M that contains more valuable information.

Regarding short-term OD forecasting, previous studies typically used OD flows from the last several time slots as model inputs to forecast the OD flow in the subsequent time interval [20, 37]. However, obtaining real-time URT OD flow is not feasible due to travel duration. As a result, these approaches cannot be applied to real-time OD predictions in URT. However, real-time inflow/outflow data are available. Therefore, in this study, we propose an input control-gated mechanism that determines the range of OD data inputs and combines it with real-time inflow/outflow information. This allows us to make more accurate and real-time predictions for URT OD flows.

To address the need for a stable and flexible forecasting model that can effectively handle spatial-temporal features while considering the unique characteristics of the URT system, we propose an enhanced spatial-temporal LSTM (ST-LSTM). This model aims to predict the passenger volume between OD pairs. The inputs to the model include historical data on OD flow, inflow/outflow, physical distance, and relevant operation data from the URT network. The output is an estimated OD passenger flow volume for the specific OD pair under consideration. The symbols used in our model are shown in Table 2.

Methodologies

Recurrent neural network and the long short-term memory

The recurrent neural network (RNN) is a type of neural network commonly used for processing sequential data. It takes sequential data as input, connects the current output with the previous input/output, and retains a certain memory to store processed information. Figure 1 illustrates the basic structure of an RNN.

In Fig. 1, X_t represents the input at time step t . h_t is the hidden layer state at time step t , which serves as the “memory” of the network. O_t denotes the output at time step t . The input, transfer, and output weights are represented by U , W , and V , respectively. It can be observed that the

RNN structure is repetitive, with shared weights. While the RNN adds connections among the hidden layers to provide memory to the neural network, it suffers from issues such as gradient disappearance and explosion. These problems make it challenging to associate forecast information with related information that exceeds a certain distance.

To overcome these challenges, Hochreiter and Schmidhuber introduced the LSTM (long short-term memory) network, which can effectively handle and forecast events with relatively long time-series intervals or delays [7]. The basic structure of LSTM is illustrated in Fig. 2. The main difference between LSTM and RNN is the addition of a structure for determining the relevance of information. This structure is called a cell and consists of three gates: the forget gate, the input gate, and the output gate. Additionally, LSTM incorporates a cell state C to retain long-term information. The working principle of LSTM can be divided into four steps:

$$\tilde{C}_t = \tanh(W_c[h_{t-1}, X_t] + b_c), \quad (7)$$

where W_i and b_i represent the input gate’s weight matrix and bias term, respectively. Similarly, W_c and b_c represent the calculated cell state’s weight matrix and bias term, respectively.

Step 1: When new information X_t is input into the LSTM, the forget gate determines which old information should be discarded. This process can be expressed using Eq. (8):

$$f_t = \sigma(W_f[h_{t-1}, X_t] + b_f), \quad (8)$$

where W_f and b_f represent the forget gate’s weight matrix and bias term, respectively. $[h_{t-1}, X_t]$ is a vector formed by combining h_{t-1} and X_t , and σ represents the sigmoid function.

Step 2: The input gate determines the extent to which the current input X_t is retained in the current cell state C_t to avoid storing irrelevant information. This process involves two tasks:

- (i) The sigmoid layer determines the amount of information that needs to be retained:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i). \quad (9)$$

- (ii) Tanh generates the candidate \tilde{C}_t :

Step 3: Combine the above two steps to update the old cell state C_{t-1} . The sigmoid function selects the update content, and the tanh function generates the candidate for the update. The new cell state C_t is obtained by discarding insignificant information and incorporating new information, as shown in Eq. (10):

Table 2 Symbols explanation in the model

Notations	Definitions
<i>Variables</i>	
$m_{ij}^{d,t}$	The OD flow from station i to station j in the time slot t on day d
$f_i^{d,t}$	The inflow/outflow entering/exiting station i in the time slot t on day d
m_{ij}^u	The total ridership from station i to station j on day u
w_{ij}	The travel time limit from station i to station j , indicating that all the passengers complete their journey from station i to station j within w_{ij} time slots
$f_{s(\text{inflow})}$	The total inbound ridership from station s in the training set
$f_{e(\text{outflow})}$	The total outbound ridership from station e in the training set
m_{ij}	The total OD flow from station i to station j in the training set
f_i	The total ridership at station i in the training set
d_{ij}	The geographical distance between station i and station j
i_{ij}	The interaction strength between station i and station j
p_{ij}	The trend correlation between OD i - j and the OD to be predicted (OD s - e)
q_{ij}	The ridership contribution of OD i - j to OD s - e
r_{ij}	The location correlation between OD i - j and OD s - e
v_{ij}	The variable obtained after performing max-min normalization on the variable q_{ij}
g_{ij}	The variable obtained after performing max-min normalization on the variable r_{ij}
z_{ij}	The ultimate measure of spatial correlation between OD i - j and OD s - e
<i>Sets</i>	
M_{ij}^d	The set of OD flows from station i to station j in different slots on day d
F_i^d	The set of inflows/outflows entering/exiting station i in different slots on day d
M_{ij}	The set of OD flow between OD i - j in the training set
C_k	The set of operating days belonging to class k
L	The set of all class l operating days before day d
$H_{O_x}^L$	The set of historical ridership for OD O_x on the days belonging to set L
$F_{s(\text{inflow})}^L$	The set of inbound ridership from station s on the days belonging to set L
$F_{e(\text{outflow})}^L$	The set of outbound ridership from station e on the days belonging to set L
<i>Matrices</i>	
M^u	The OD matrix on day u
W	The matrix of travel time limit between ODs
<i>Parameters</i>	
n	The number of stations in urban rail transit
H	The number of time slots that one day is divided into
h	The number of time slots corresponding to the real-time data used for model input
u	The number of days for which the training set contains data
k	The number of classifications obtained by clustering operational days
a	The number of days contained in set L
x	The number of ODs that have a higher spatial correlation with OD s - e

$$C_t = f \odot C_{t-1} + i_t \odot \tilde{C}_t, \tag{10}$$

$$h_t = O_t \odot \tanh(C_t), \tag{12}$$

where \odot represents multiplication by elements.

Step 4: The output gate determines the current output value h_t , which is related to the current cell state C_t . Equations (11) and (12) express this process:

$$O_t = \sigma(W_o[h_{t-1}, X_t] + b_o), \tag{11}$$

where W_o and b_o are the output gate's weight matrix and bias term, respectively.

Fig. 1 Recurrent neural network (RNN) structure

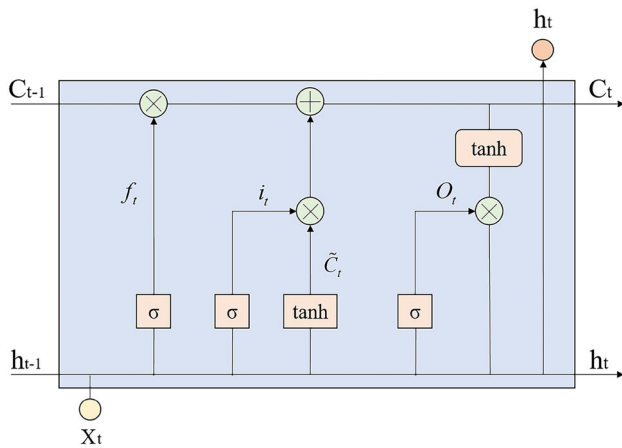
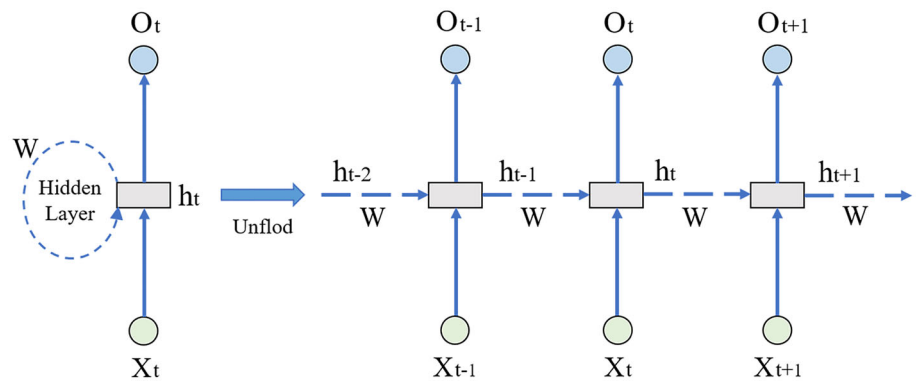


Fig. 2 The basic structure of LSTM

Model development

Typically, LSTM models use a vector that represents the historical data of the OD to be predicted as input. However, when applying LSTM to predict OD flow, only the data from a single OD are considered, disregarding the interactions between ODs. This approach fails to account for the complex network structure of URT systems in major cities like Beijing, where certain ODs exhibit close connections regarding passenger flows. Consequently, LSTM has inherent limitations. In contrast, our proposed model takes two-dimensional matrices as inputs, representing historical flows across different ODs and time periods. This approach captures the OD flow data’s spatial and temporal features.

Figure 3 illustrates the framework of our proposed prediction method based on the improved long short-term memory (ST-LSTM). The framework consists of four main components: the temporal feature extraction module, the spatial correlation learning module, the data inflow control gate, and the fusion and prediction module. The temporal feature extraction module aims to cluster the operating days and extract the travel time between origin–destination (OD) pairs, which is used to determine the time range of the input data for

subsequent modules. The spatial correlation learning module measures the spatial correlation between OD pairs using three indicators and filters ODs with a stronger correlation to the OD pair being predicted. The data inflow control gate takes time information and real-time data as inputs and outputs available real-time data through the data control mechanism. The fusion module receives processed historical and real-time data and combines them into a two-dimensional matrix for the prediction module. The output of this module is the predicted OD flow for the OD pair being predicted. In the following subsections, we elaborate on these main modules.

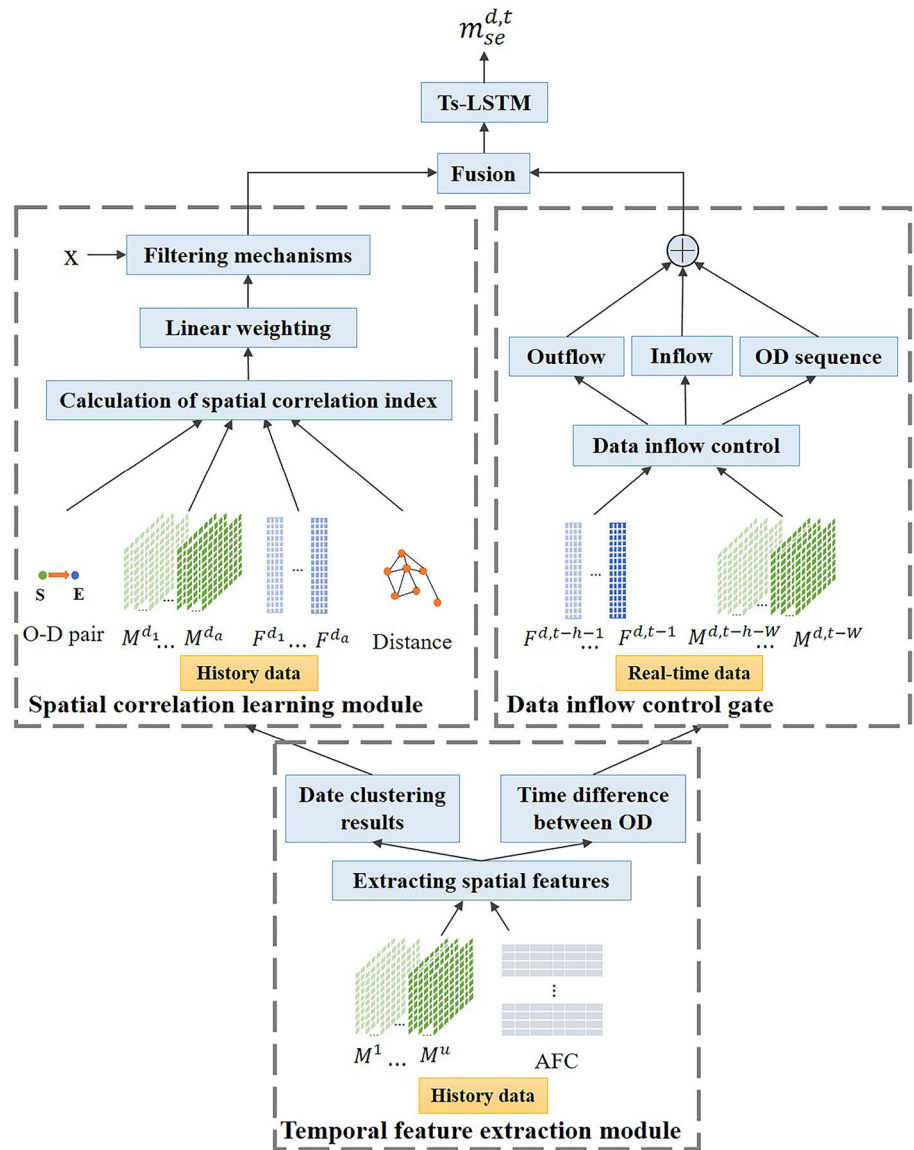
Temporal feature extraction module

Time feature pertains to attributes used to describe temporal changes. Specifically, this module focuses on extracting two attributes: date and travel time. Typically, previous works utilize extensive continuous historical data to predict OD flow [1, 13]. However, these approaches often include irrelevant information from operating days with different travel patterns and fail to incorporate real-time data in OD predictions. To overcome these challenges, we have designed the temporal feature extraction module. It aims to filter out more effective training data and determine the range of available real-time data for improved prediction accuracy.

To identify more effective training data, we classify operating days based on their daily travel patterns. First, we extract the network-wide origin–destination (OD) matrix from the training set data, which includes u days with a complete operating day (i.e., 24 h) as the time interval. This results in obtaining u matrices of size $n \times n$. The expression for the OD matrix on the day u is given by Eq. (13).

$$M^u = \begin{bmatrix} m_{11}^u & m_{12}^u & m_{13}^u & \dots & m_{1n}^u \\ m_{21}^u & m_{22}^u & m_{23}^u & \dots & m_{2n}^u \\ m_{31}^u & m_{32}^u & m_{33}^u & \dots & m_{3n}^u \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n1}^u & m_{n2}^u & m_{n3}^u & \dots & m_{nn}^u \end{bmatrix}_{n \times n}, \quad (13)$$

Fig. 3 The overview of ST-LSTM framework



where m_{ij}^u is total ridership from station i to station j on day u . (14)

To implement the clustering algorithm, we begin by deleting the diagonal elements of each matrix. Then, we transform each matrix into an $(n \times (n - 1))$ column vector, where n represents the total number of stations and $(n \times (n - 1))$ is the total number of ODs. Next, we combine the network-wide OD flows for u days into a matrix of size $(n(n - 1) \times u)$, denoted as $OdData$, which is expressed in Eq. (14).

$$OdData = \begin{bmatrix} m_{12}^1 & m_{12}^2 & m_{12}^3 & \dots & m_{12}^u \\ m_{13}^1 & m_{13}^2 & m_{13}^3 & \dots & m_{13}^u \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{ij}^1 & m_{ij}^2 & m_{ij}^3 & \dots & m_{ij}^u \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ m_{n,n-1}^1 & m_{n,n-1}^2 & m_{n,n-1}^3 & \dots & m_{n,n-1}^u \end{bmatrix}_{n(n-1) \times u} \quad i \neq j,$$

Finally, we apply an improved K-means clustering algorithm based on the elbow method [9] to cluster the matrix $OdData$. This method automatically determines the optimal number of clusters, denoted as k , and classifies the operating days into k classes. The resulting clustering results are represented by the symbol C :

$$C = \{C_1, C_2, \dots, C_k\}, \tag{15}$$

where C_k denotes the set of operating days belonging to class k .

As mentioned earlier, obtaining the OD matrix in the last few time intervals is impossible in URT (real-time urban traffic). Therefore, it is crucial to determine the range of available

real-time OD data. To tackle this challenge, we introduce a parameter called OD travel time limit (W), which draws inspiration from the concept of the 85th percentile speed commonly used as the limit speed in traffic scenarios.

The 85th percentile speed is the speed below which 85% of the vehicles on the road are traveling and is commonly used as the maximum speed limit [26]. To study the distribution pattern of OD travel time, we present the cumulative frequency distribution curve of OD travel time using one month of Beijing Metro data, as shown in Fig. 4. From Fig. 4a, it can be observed that the growth trend of the cumulative frequency distribution curve significantly slows down when the ordinate value reaches 95%. This indicates that 95% of passengers' travel time between this OD falls within 16 min (corresponding to the horizontal value), while the travel time for others is longer and more uneven. Figure 4b illustrates the cumulative frequency distribution curves of travel time for 30 randomly selected ODs. It can be observed that the curves exhibit a similar pattern for different ODs. Therefore, we determine the travel time limit by dividing the horizontal coordinate value of the inflection point on the cumulative frequency distribution curve of each OD (i.e., the corresponding OD travel time) by the time span and rounding it up to an integer. This paper defines the travel time limit as the maximum allowable travel time for passengers traveling between OD pairs. For example, in the case of Beijing Metro mentioned above, W corresponds to the 95th percentile travel time for each OD. The OD travel time limit (W) is expressed by Eq. (16).

$$W = \begin{bmatrix} w_{11} & w_{12} & w_{13} & \dots & w_{1n} \\ w_{21} & w_{22} & w_{23} & \dots & w_{2n} \\ w_{31} & w_{32} & w_{33} & \dots & w_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ w_{n1} & w_{n2} & w_{n3} & \dots & w_{nn} \end{bmatrix}_{n \times n}, \quad (16)$$

where w_{ij} represents the travel time limit from station i to station j , indicating that all the passengers complete their journey from station i to station j within w_{ij} time slots.

Spatial correlation learning module

Previous studies have made assumptions regarding the influence of passenger flow between OD pairs and have proposed different approaches to capture these influences. Some studies utilize Convolutional Neural Networks (CNNs) to capture global spatial influences over the entire network [3, 12]. Others focus on local influences, considering OD pairs with origins and destinations in close geographical proximity [13, 46]. Different from these existing works, we introduce a comprehensive indicator that measures the spatial correlation between OD pairs across the entire network. Our approach

considers both the travel pattern and geographical location to provide a more comprehensive understanding of the spatial correlations between OD pairs.

In our proposed method, we introduce three indicators, namely p , q , and r , to measure the spatial correlations based on passenger flow data, OD matrix, and geographical data of the rail network. Let us consider OD s - e as the OD pair to be predicted. First, we use p_{ij} to represent the trend correlation between OD i - j and OD s - e . The expression for p_{ij} , derived using the Pearson Correlation Coefficient, is shown in Eq. (17).

$$p_{ij} = \left| \frac{Cov(M_{se}, M_{ij})}{\sigma(M_{se})\sigma(M_{ij})} \right|, \quad (17)$$

where M_{se} represents the sequence of all OD flow used for analysis for training sets between OD s - e . Cov denotes the covariance function, and σ represents the standard deviation calculation function. The value range of p_{ij} is $[0, 1]$, where a larger value indicates a stronger trend correlation between the OD pairs.

Let us consider $f_{s(\text{inflow})}$ as the total inbound ridership from station s and $f_{e(\text{outflow})}$ as the total outbound ridership from station e in the training set, as shown in Eq. (18).

$$f_{s(\text{inflow})} = \sum_{d=1}^u \sum_{j=1}^n m_{sj}^d, \quad (18a)$$

$$f_{e(\text{outflow})} = \sum_{d=1}^u \sum_{i=1}^n m_{ie}^d. \quad (18b)$$

Here, q_{ij} represents the ridership contribution of OD i - j to OD s - e , as shown in Eq. (19).

$$q_{ij} = \frac{m_{ie}}{f_{e(\text{outflow})}} + \frac{m_{sj}}{f_{s(\text{inflow})}},$$

$$m_{ie} = \sum_{d=1}^u m_{ie}^d; \quad m_{sj} = \sum_{d=1}^u m_{sj}^d, \quad (19)$$

where m_{sj}^d represents the total ridership from station s to station j on day d , m_{sj} is the total OD flow from station s to station j in the training set.

We define the maximum value as q_{\max} and the minimum value as q_{\min} . ODs are weakly correlated if both their origins and destinations are far apart [1]. Inspired by the distance decay theory [27, 38], indicator i is used to express the interaction strength between stations, as shown in Eq. (20):

$$i_{ij} = \frac{f_i f_j}{d_{ij}^2}, \quad f_i = f_{i(\text{outflow})} + f_{i(\text{inflow})};$$

$$f_j = f_{j(\text{outflow})} + f_{j(\text{inflow})}, \quad (20)$$

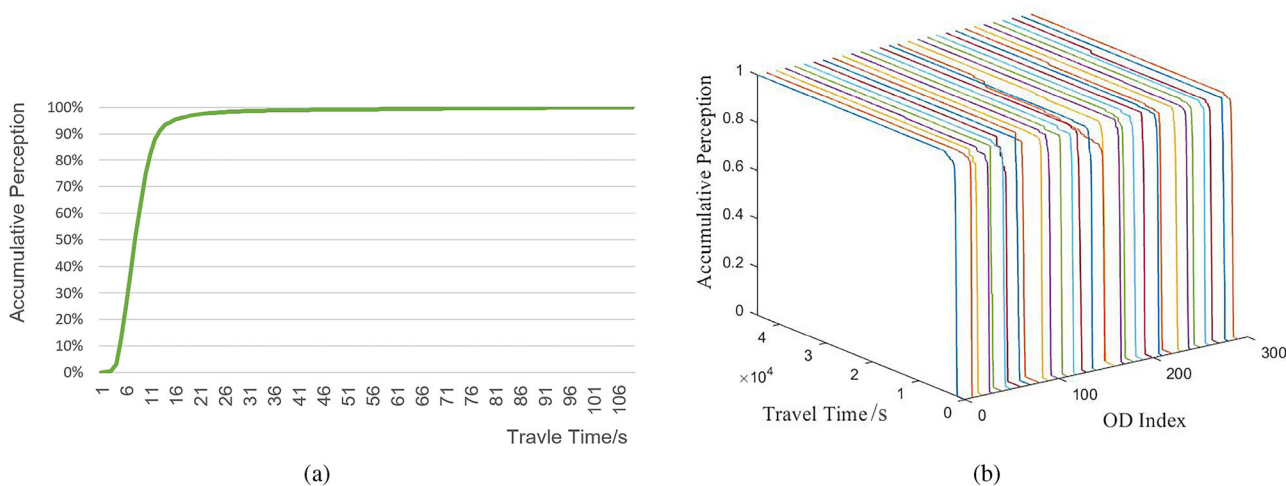


Fig. 4 Cumulative frequency distribution curves of OD travel time. **a, b** shows the curves for single OD and 30 ODs respectively

where i_{ij} represents the interaction strength between station i and station j . d_{ij} is the geographical distance between station i and station j , and f_j denotes the total ridership at station j . Additionally, r_{ij} represents the location correlation between OD i - j and OD s - e , as shown in Eq. (21).

$$r_{ij} = \frac{1}{4}(i_{rs} + i_{je} + i_{ie} + i_{js}). \tag{21}$$

We define the maximum value as r_{\max} and the minimum value as r_{\min} . To ensure that indicator q and indicator r are on a common scale, we normalize them using the following two equations:

$$v_{ij} = \frac{q_{\max} - q_{ij}}{q_{\max} - q_{\min}}, \tag{22a}$$

$$g_{ij} = \frac{r_{\max} - r_{ij}}{r_{\max} - r_{\min}}, \tag{22b}$$

where q_{\max} , q_{\min} , r_{\max} , and r_{\min} represent the maximum and minimum values of the two indicators. v_{ij} and g_{ij} denote the normalized values of the two indicators. Next, we transform the three indicators into one using the linear weighted compromise method. The final indicator is expressed in Eq. (23).

$$z_{ij} = \omega_1 \cdot p_{ij} + \omega_2 \cdot v_{ij} + \omega_3 \cdot g_{ij}. \tag{23}$$

After sorting z_{ij} and selecting the top x OD pairs as O_1, O_2, \dots, O_x , it becomes evident that these top x OD pairs exhibit the highest spatial correlation with the OD pair to be predicted. Additionally, there exist strong correlations between inflows/outflows and OD flows, as shown in Eqs. (1)–(5). Taking this relationship into account, the outputs of the spatial correlation learning module include the strongly correlated OD pairs O_1, O_2, \dots, O_x , as well as stations s and e .

Data inflow control gate

As discussed in the previous section, obtaining the OD matrix in the last few time intervals is impossible due to travel time constraints. Therefore, it is crucial to consider how to incorporate real-time data into OD predictions. The OD travel time limit (W), obtained from the temporal feature extraction module, is utilized by the data inflow control gate to determine the range of available OD data. This allows us to obtain the available real-time OD flow, as shown in Eq. (24).

$$M_{ij}^d = \left\{ m_{ij}^{d,t-w_{ij}-h}, m_{ij}^{d,t-w_{ij}-h+1}, \dots, m_{ij}^{d,t-w_{ij}-1}, m_{ij}^{d,t-w_{ij}} \right\}, \tag{24}$$

where w_{ij} represents the travel time limit for passengers from station i to station j , and $m_{ij}^{d,t-w_{ij}}$ represents the OD flow from station i to station j in the time slot x on day d . Based on the discussion in “Temporal feature extraction module” section, this paper assumes that all passengers complete their trip from station i to station j within w_{ij} time slots. This means all passengers have finished their journey within the $t - w_{ij}$ time slot and the time slot before, so the OD flow data for these time slots are complete and available. It is important to note that, to maintain consistency in the dimensions of the input data, the data selected for day d do not start from the first time slot but instead select the latest h time slot data for input.

Moreover, motivated by the relationship between inflows/outflows and OD flows, we simultaneously incorporate the real-time inflows/outflows, as shown in Eq. (25). This fusion allows us to combine the inflow/outflow data with the available real-time OD data for further processing in the next

module.

$$F_{s(\text{outflow})}^d = \left\{ f_s^{d,t-h-1}, f_s^{d,t-h}, \dots, f_s^{d,t-2}, f_s^{d,t-1} \right\}, \tag{25a}$$

$$F_{e(\text{inflow})}^d = \left\{ f_e^{d,t-h-1}, f_e^{d,t-h}, \dots, f_e^{d,t-2}, f_e^{d,t-1} \right\}. \tag{25b}$$

Fusion and prediction module

After obtaining the spatial–temporal features of the historical data and available real-time data from the previous modules, it is necessary to integrate these datasets before making predictions.

First, we fuse historical data based on their spatial–temporal features. As mentioned earlier, the temporal feature extraction module classifies the operating days into k classes, while the spatial correlation learning module outputs the first x OD pairs with high spatial correlation, as well as the origin and destination of the OD pair, to be predicted. Let us assume that the day to be predicted (denoted as d) belongs to the class l operating days, i.e., $d \in C_l$ (C_l represents the set of operating days belonging to class l). For convenience, let $L = \{d_1, d_2, d_3, \dots, d_a\}$ represent the set of all class l operating days before day d , where the subscript denotes the sequential number in the set. The fused historical data are then obtained using Eq. (26):

$$H_{O_1}^L = \left\{ M_{O_1}^{d_1}, M_{O_1}^{d_2}, \dots, M_{O_1}^{d_a} \right\}, \tag{26a}$$

$$H_{O_2}^L = \left\{ M_{O_2}^{d_1}, M_{O_2}^{d_2}, \dots, M_{O_2}^{d_a} \right\}, \tag{26b}$$

$$H_{O_x}^L = \left\{ M_{O_x}^{d_1}, M_{O_x}^{d_2}, \dots, M_{O_x}^{d_a} \right\}, \tag{26c}$$

$$H_{se}^L = \left\{ M_{se}^{d_1}, M_{se}^{d_2}, \dots, M_{se}^{d_a} \right\}, \tag{26d}$$

$$F_{s(\text{outflow})}^L = \left\{ F_s^{d_1}, F_s^{d_2}, \dots, F_s^{d_a} \right\}, \tag{26e}$$

$$F_{e(\text{inflow})}^L = \left\{ F_e^{d_1}, F_e^{d_2}, \dots, F_e^{d_a} \right\}, \tag{26f}$$

where $H_{O_x}^L$ represents the set of historical ridership for OD O_x on the days belonging to set L , which contains a elements. $M_{O_x}^{d_a}$ represents the ridership sequence of OD O_x on day d_a which contains H elements, and $F_{s(\text{outflow})}^L$ denotes the set of outbound ridership from station s on the days belonging to set L .

Next, we process the real-time data obtained from the data inflow control gate, which needs to be converted into an array with the same number of dimensions as Eq. (26). The real-time data array is obtained by combining the output of the

spatial correlation learning module, as shown in Eq. (27).

$$M_{O_1}^d = \left\{ m_{O_1}^{d,t-w_{O_1}-h}, m_{O_1}^{d,t-w_{O_1}-h+1}, \dots, m_{O_1}^{d,t-w_{O_1}-1}, m_{O_1}^{d,t-w_{O_1}} \right\}, \tag{27a}$$

$$M_{O_2}^d = \left\{ m_{O_2}^{d,t-w_{O_2}-h}, m_{O_2}^{d,t-w_{O_2}-h+1}, \dots, m_{O_2}^{d,t-w_{O_2}-1}, m_{O_2}^{d,t-w_{O_2}} \right\}, \tag{27b}$$

$$M_{O_x}^d = \left\{ m_{O_x}^{d,t-w_{O_x}-h}, m_{O_x}^{d,t-w_{O_x}-h+1}, \dots, m_{O_x}^{d,t-w_{O_x}-1}, m_{O_x}^{d,t-w_{O_x}} \right\}, \tag{27c}$$

$$M_{se}^d = \left\{ m_{se}^{d,t-w_{se}-h}, m_{se}^{d,t-w_{se}-h+1}, \dots, m_{se}^{d,t-w_{se}-1}, m_{se}^{d,t-w_{se}} \right\}, \tag{27d}$$

$$F_{s(\text{outflow})}^d = \left\{ f_s^{d,t-h-1}, f_s^{d,t-h}, \dots, f_s^{d,t-2}, f_s^{d,t-1} \right\}, \tag{27e}$$

$$F_{e(\text{inflow})}^d = \left\{ f_e^{d,t-h-1}, f_e^{d,t-h}, \dots, f_e^{d,t-2}, f_e^{d,t-1} \right\}. \tag{27f}$$

Finally, the historical data and real-time data are fused to obtain a two-dimensional matrix, as shown in Eq. (28). This matrix is then passed into the prediction module, and the output is the predicted ridership for OD pair s – e .

$$Input = \begin{bmatrix} H_{O_1}^L & M_{O_1}^d \\ H_{O_2}^L & M_{O_2}^d \\ \vdots & \vdots \\ H_{O_x}^L & M_{O_x}^d \\ H_{se}^L & M_{se}^d \\ F_{s(\text{outflow})}^L & F_{s(\text{outflow})}^d \\ F_{e(\text{inflow})}^L & F_{e(\text{inflow})}^d \end{bmatrix}_{(x+3)(aH+h)}. \tag{28}$$

Optimization and training

The output of the model is the predicted OD flow. During the training process, we aim to minimize the error between the predicted OD flow and the actual OD flow. To achieve this, we employ the mean squared error (MSE) as the loss function, which can be written as follows:

$$L(\theta) = \frac{1}{n} \sum_{i=0}^{n-1} (m^{t+i} - \hat{m}^{t+i})^2, \tag{29}$$

where \hat{m}^{t+i} represents the predicted OD flow, while m^{t+i} represents the actual OD flow. The parameter θ encompasses all the learnable parameters in our network, which can be obtained through the back-propagation algorithm and Adam optimizer [14].

Table 3 Data description

Description	MetroNJ2018	MetroBJ2019
Data	March 1, 2018–March 31, 2018	September 1, 2019–September 30, 2019
Time	05:00–23:00	05:00 to 23:00
Data record	52 million	138 million
Station number	175	351
OD number	175 × 175	351 × 351
Time interval	15 min	15 min
Timestamp number in a day	72	72

Experiment

In this section, we perform comprehensive experiments on two real-world datasets to evaluate our proposed model and compare it with benchmark methods. The experimental results are further analyzed from various perspectives.

Data description

Two datasets, namely the Nanjing Subway and Beijing Subway, were used in the experiments, as shown in Table 3. The Nanjing dataset (MetroNJ2018) contains over 50 million records from 175 stations in March 2018 in Nanjing, China. Similarly, the Beijing dataset (MetroBJ2019) contains over 130 million records from 351 stations in September 2019 in Beijing, China. Each record includes information such as the card number, entry-station name, exit-station name, entry time, and exit time. Inflows/outflows and OD flows are extracted every 15 min based on Eqs. (1) and (3). A single operating day consists of 72 time periods, starting from 5:00 and ending at 23:00. To ensure standardization, all data is normalized using the min–max scaler.

Due to the limitations of computational resources, we encountered difficulties in evaluating the model using the OD flows of the entire network. Therefore, we employed stratified sampling to select a subset of OD samples for analysis. The OD flows were classified into three levels (high, medium, and low) based on their ridership. We extracted 20% of OD samples from each level for further analysis. We used the last seven days of data as testing data, while the remaining data was utilized for training.

Experiment settings

To expedite the learning and convergence during model training, we perform min–max normalization to scale the data within the range of [0, 1], as depicted in Eq. (30).

$$z = \frac{q - \min(q)}{\max(q) - \min(q)}. \quad (30)$$

Based on parameter tuning results, our final model consists of two hidden layers with a hidden size of 64 for each layer. We utilize the Adam optimizer during the training process with a learning rate 0.0005 and a batch size of 12. As for the comparison models, we carefully tuned them and selected the best parameters. Our model is implemented using PyTorch 1.1.0, and all deep learning models were executed on an NVIDIA GeForce RTX 3060 Laptop GPU.

In this study, we have selected three commonly used metrics to evaluate the performance of our model: root mean square error (RMSE), mean average error (MAE), and mean absolute percentage error (MAPE).

$$\text{RMSE} = \sqrt{\frac{1}{\varepsilon} \sum_{i=1}^{\varepsilon} (m_i - \widehat{m}_i)^2}, \quad (31)$$

$$\text{MAE} = \frac{1}{\varepsilon} \sum_{i=1}^{\varepsilon} \|m_i - \widehat{m}_i\|, \quad (32)$$

$$\text{MAPE} = \frac{1}{\varepsilon} \sum_{i=1}^{\varepsilon} \left\| \frac{m_i - \widehat{m}_i}{m_i} \right\|, \quad (33)$$

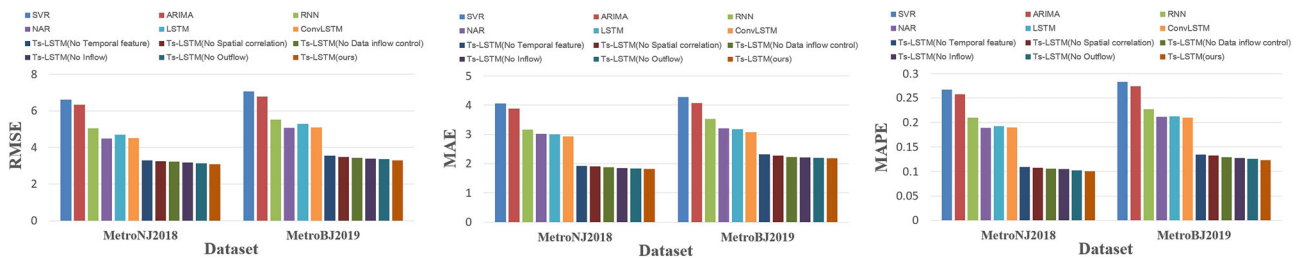
where ε represents the total number of predicted values, m_i refers to the actual OD ridership, and \widehat{m}_i denotes the predicted OD ridership.

Methods for comparison

To evaluate the performance of our model, we compare it with several other models, namely LSTM, ARIMA, SVR, RNN, ConvLSTM, and NAR. Additionally, we construct five additional models based on our approach to showcase the effectiveness of the proposed temporal feature extraction module, spatial correlation learning module, and data inflow control mechanism. For all these models, apart from the control component, all other parameters remain the same as those in ST-LSTM. The detailed information for the comparison methods is listed below:

Table 4 Comparison of performances of different models

Models	MetroNJ2018			MetroBJ2019		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.595	4.051	0.2669	7.068	4.269	0.2828
ARIMA	6.329	3.883	0.2575	6.777	4.075	0.2739
RNN	5.026	3.158	0.2093	5.516	3.527	0.2271
NAR	4.480	3.011	0.1884	5.049	3.206	0.2109
LSTM	4.677	2.999	0.1926	5.298	3.181	0.2124
ConvLSTM	4.499	2.926	0.1898	5.083	3.068	0.2098
ST-LSTM (no temporal feature)	3.281	1.913	0.1090	3.539	2.321	0.1346
ST-LSTM (no spatial correlation)	3.236	1.903	0.1071	3.462	2.282	0.1323
ST-LSTM (no data inflow control)	3.217	1.877	0.1060	3.417	2.228	0.1289
ST-LSTM (no inflow)	3.157	1.851	0.1044	3.362	2.213	0.1273
ST-LSTM (no outflow)	3.110*	1.834*	0.1022*	3.338*	2.198*	0.1259*
ST-LSTM (ours)	3.070	1.812	0.1006	3.284	2.175	0.1233
Improvements	1.31%	1.19%	1.50%	1.61%	1.05%	2.08%

**Fig. 5** Comparison of performances of different models

- LSTM [39, 56]: This model is a recurrent neural network designed for handling sequential data. It consists of two hidden layers, and the hidden sizes for both layers are set as 64. The learning rate is set to 0.0005, and the batch size is 12.
- ARIMA [4]: This model is a mathematical modeling approach for time-series prediction. Based on parameter tuning results, the optimal model selected for ARIMA is (2, 1, 0).
- SVR [8]: Support vector regression is a regression forecasting model. Here, we complete the experiment with the SVR with the RBF kernel.
- RNN [25]: The recurrent neural network is a deep learning method that analyzes sequential data. The hidden size is 64, the learning rate is 0.0005, and the batch size is 12.
- ConvLSTM [29]: Convolutional LSTM is a deep learning method designed explicitly for spatial–temporal analysis. It replaces the fully connected layer in LSTM with a convolutional structure. There are two layers with 8 and 1 filters, respectively. The kernel size is 3×3 . The learning rate is 0.0005. The batch size is 12.
- NAR [16, 41]: Nonlinear AutoRegressive is a time series prediction model designed to capture complex patterns and dynamic features in time-series data. The NAR model

utilizes a neural network as the nonlinear function, with two hidden layers and 64 neurons in each layer. The delay order is set to 10.

- ST-LSTM (No temporal feature): We remove the temporal feature extraction module to assess its impact on the model’s effectiveness. The input is the historical data.
- ST-LSTM (No spatial correlation): We remove the spatial correlation learning module to verify its effectiveness. In this case, the input consists of the data of the OD pairs to be predicted.
- ST-LSTM (No data inflow control): We remove the data inflow control mechanism to demonstrate its impact on the model’s effectiveness.
- ST-LSTM (No inflow): We remove the inbound passenger flow branch to evaluate the effectiveness of the inflow-gated mechanism.
- ST-LSTM (No outflow): We remove the outbound passenger flow branch to evaluate the effectiveness of the outflow-gated mechanism.
- ST-LSTM: The whole model we propose in “[Methodologies](#)” section.

Experimental results

Overall performance

The experimental results are shown in Table 4 and Fig. 5, and the best results for each evaluation metric are highlighted in bold.

Based on Table 4 and Fig. 5, it can be observed that non-deep learning methods such as SVR and ARIMA exhibit weaker performance compared to deep learning methods across all evaluation metrics. This could be attributed to the limited ability of non-deep learning methods to capture the complex spatial–temporal features involved in OD flow estimation. Moreover, our ST-LSTM model achieves the best performance among the compared models. Specifically, on the MetroNJ2018 dataset, our model outperforms other models by at least 1.31%, 1.19%, and 1.50% in terms of RMSE, MAE, and MAPE, respectively. On the MetroBJ2019 dataset, the improvement ratios for RMSE, MAE, and MAPE are at least 1.61%, 1.05%, and 2.08%, respectively. These results further validate the effectiveness of our proposed ST-LSTM model for OD flow estimation.

The proposed temporal feature extraction module, spatial correlation learning module, and data inflow control mechanism are proven effective in improving model performance. When comparing ST-LSTM (no temporal feature) to TS-LSTM, the RMSE is improved by 6.44% for MetroNJ2018 and 7.20% for MetroBJ2019. Similarly, when comparing ST-LSTM (no spatial correlation) to ST-LSTM, the RMSE is improved by 5.16% for MetroNJ2018 and 5.13% for MetroBJ2019. Additionally, when comparing ST-LSTM (no data inflow control) to ST-LSTM, the RMSE is improved by 4.59% for MetroNJ2018 and 3.89% for MetroBJ2019. Regardless of the case, ST-LSTM consistently performs the best, benefiting from its architecture incorporating the temporal feature extraction module, spatial correlation learning module, and data inflow control mechanism.

To determine whether there is a significant performance difference between the proposed model and the benchmark

Table 6 The performance of ST-LSTM at different time spans on MetroNJ2018

Time span	RMSE	MAE	MAPE	Time requires
5min	4.790	3.066	0.1578	332.076s
10min	3.205	2.079	0.1053	198.740s
15min	3.070	1.812	0.1006	158.432s
30min	5.661	3.341	0.1856	97.586s
60min	9.596	5.664	0.1931	61.353s

models, we conducted a *t*-test to compare the prediction error RMSE of the models at different periods. Table 5 presents the statistics (*t*-value) and significance index (*p*-value) between the ST-LSTM and benchmark models.

Table 5 shows that the *p*-values corresponding to the benchmark models are less than 0.05 (given the significance level $\alpha = 0.05$) for all data sets. This indicates a significant difference between the ST-LSTM and the benchmark model, thereby validating the effectiveness and reliability of the ST-LSTM model.

Prediction performances of individual OD pairs

To evaluate the performance of prediction models on individual OD flows, we selected several OD flows and compared the actual ridership with the predicted ridership, as shown in Figs. 6 and 7. As can be seen, OD_1 and OD_4 are characterized by peak flows during evening hours. Whether it is MetroNJ2018 or MetroBJ2019, the ST-LSTM model accurately captures the variations throughout the day and even accurately predicts the peak flows. For OD_2 and OD_5, which exhibit peak features during morning hours and variations during evening hours, the ST-LSTM model accurately captures the overall trend despite certain variations. Furthermore, for OD_3 and OD_6, both of which represent small OD flows without peak features but significant variations throughout the day, the ST-LSTM model performs well, as depicted in Fig. 7. In conclusion, our proposed ST-LSTM model demonstrates good performance at an individual level for most cases in these two real-world metro datasets.

Performance under different temporal scenarios

We have demonstrated that ST-LSTM outperforms other models in various spatial scenarios. Next, we evaluate our proposed model under different temporal scenarios. First, we carefully choose the prediction time span to evaluate our model. It is common in various short-term OD prediction studies to use 15 min [15, 52], 30 min [46, 53], or 60 min [1, 13] as the prediction time span. This selection allows for easy comparison and benchmarking with existing research findings. Additionally, the choice of time span aims

Table 5 The *t*-test results between the ST-LSTM model and the benchmark models

Models	MetroNJ2018		MetroBJ2019	
	<i>t</i> -value	<i>p</i> -value	<i>t</i> -value	<i>p</i> -value
SVR	4.047	0.000	5.116	0.000
ARIMA	4.061	0.000	4.461	0.000
RNN	3.491	0.001	3.874	0.000
NAR	2.817	0.004	2.608	0.009
LSTM	2.782	0.005	2.930	0.003
ConvLSTM	2.237	0.026	2.322	0.020

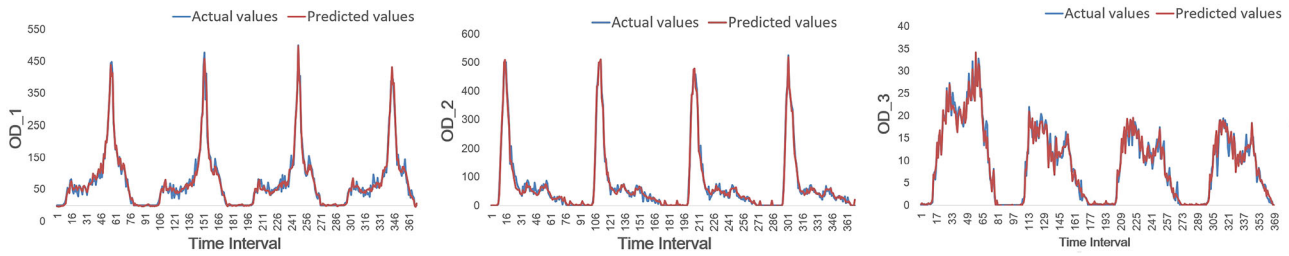


Fig. 6 The actual and predicted flows comparison of selected OD pairs in MetroNJ2018

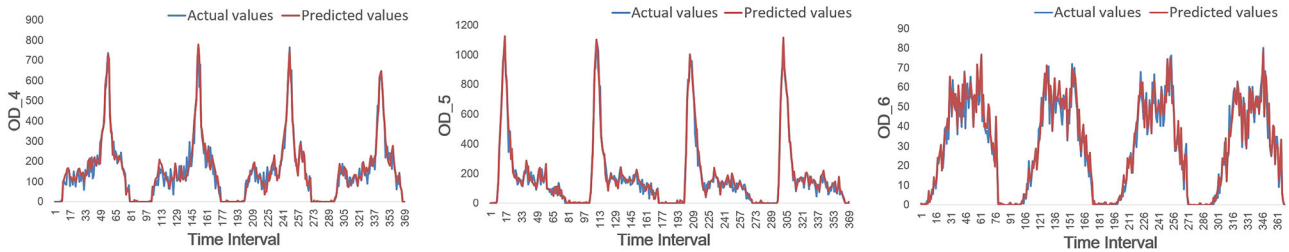


Fig. 7 The actual and predicted flows comparison of selected OD pairs in MetroBJ2019

Table 7 The performance of prediction models at different time steps on MetroNJ2018

Models	15 min			30 min			60 min		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.595	4.051	0.2669	11.982	7.361	0.3600	14.732	9.050	0.3826
ARIMA	6.329	3.883	0.2575	11.940	7.325	0.3579	14.511	8.902	0.3684
RNN	5.026	3.158	0.2093	9.268	5.824	0.3220	12.672	7.963	0.3453
NAR	4.480	3.011	0.1884	7.730	4.888	0.3206	12.751	7.799	0.3250
LSTM	4.677	2.999	0.1926	7.523	4.824	0.3098	12.143	7.786	0.3316
ConvLSTM	4.499	2.926	0.1898	7.188	4.707	0.3054	11.628	7.514	0.3230
ST-LSTM (no temporal feature)	3.281	1.913	0.1090	5.773	3.483	0.1917	9.649	5.676	0.2186
ST-LSTM (no spatial correlation)	3.236	1.903	0.1071	5.769	3.392	0.1909	9.633	5.665	0.2174
ST-LSTM (no data inflow control)	3.217	1.877	0.1060	5.768	3.382	0.1901	9.630	5.618	0.2150
ST-LSTM (no inflow)	3.157	1.851	0.1044	5.763	3.379	0.1892	9.618	5.640	0.2092
ST-LSTM (no outflow)	3.110	1.834	0.1022	5.742	3.378	0.1886	9.614	5.668	0.2071
ST-LSTM (ours)	3.070	1.812	0.1006	5.661	3.341	0.1856	9.596	5.664	0.1931

to balance prediction accuracy and computational efficiency. To illustrate this point, we present the prediction results of the ST-LSTM for different time spans using MetroNJ2018 in Table 6 and the best results are in bold. It is evident that the program's running time increases with decreasing time span, while the prediction accuracy reaches its highest point at a time span of 15 min. This outcome suggests that a smaller time granularity may lead to fluctuations in OD flow data, making it more challenging for the model to learn passenger flow patterns effectively. Conversely, the model performs relatively worse when the time span is set at 60 min, which could be attributed to the longer time span, resulting in a reduced volume of training data and, thus, hampering the model's learning capacity. These observations

imply the importance of selecting an appropriate time span in achieving optimal prediction accuracy while considering computational resources. We observed longer computation times and suboptimal prediction results when using a time span of 5 min or 10 min. To ensure operational efficiency for subsequent comparative experiments, we set the prediction time spans as 15 min, 30 min, and 60 min, and test and compare the models accordingly. The results are presented in Table 7 and Table 8 and the best results for each metric are in bold.

We observed that among the various models evaluated, ST-LSTM exhibited the best performance, while LSTM and NAR models showcased similar performance on this dataset. Conversely, the SVR model displayed the worst perfor-

Table 8 The performance of prediction models at different time steps on MetroBJ2019

Models	15 min			30 min			60 min		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	7.068	4.269	0.2828	12.435	7.694	0.3773	15.185	9.383	0.3952
ARIMA	6.777	4.075	0.2739	12.408	7.653	0.3737	14.979	9.231	0.3819
RNN	5.516	3.527	0.2271	9.723	6.136	0.3466	13.127	8.275	0.3559
NAR	5.049	3.206	0.2109	8.014	5.181	0.3310	12.547	8.238	0.3401
LSTM	5.298	3.181	0.2124	7.983	5.153	0.3321	12.602	8.116	0.3410
ConvLSTM	5.083	3.068	0.2098	7.657	4.972	0.3271	12.035	7.885	0.3396
ST-LSTM (no temporal feature)	3.539	2.321	0.1346	6.005	3.807	0.2196	9.881	6.029	0.2500
ST-LSTM (no spatial correlation)	3.462	2.282	0.1323	6.004	3.739	0.2224	9.868	6.022	0.2489
ST-LSTM (no data inflow control)	3.417	2.228	0.1289	5.986	3.710	0.2165	9.849	6.019	0.2414
ST-LSTM (no inflow)	3.362	2.213	0.1273	5.985	3.708	0.2140	9.839	6.019	0.2404
ST-LSTM (no outflow)	3.338	2.198	0.1259	5.984	3.706	0.2140	9.860	6.006	0.2364
ST-LSTM (ours)	3.284	2.175	0.1233	5.888	3.670	0.2117	9.823	5.993	0.2243

mance. Furthermore, we noticed a consistent trend across all prediction models, where the 15-min time span yielded the highest performance, while the 60-min time span resulted in the lowest performance. This observation can be attributed to the fact that the dataset for the 60-min time span contained fewer data points than other time spans. These findings emphasize the importance of selecting an appropriate time span. Moreover, the availability of sufficient data is crucial in enabling models to capture complex patterns accurately.

To further validate our model's performance during different time periods, we divided the two datasets into rush hours and non-rush hours based on specific time intervals. Specifically, we defined rush hours as 7:00–9:00 and 17:00–19:00 on weekdays, while all other time periods were considered non-rush hours. The experimental results are presented in Table 9 and Table 10, and the best results for each evaluation metric are highlighted in bold. Additionally, Fig. 8 illustrates the trend of average evaluation metrics for the MetroNJ2018 dataset over time. We have listed several key findings below:

- Despite the high travel demand and the unpredictability of traffic conditions during rush hours, our ST-LSTM model consistently outperforms other models. These results provide strong evidence for the stability and reliability of our proposed ST-LSTM model.
- Based on Fig. 8, we can observe that the performance of different models in various time intervals follows similar patterns to the overall performance. Notably, the ST-LSTM model consistently outperforms other models, both during rush hours and non-rush hours. These results indicate the superior adaptability of the ST-LSTM model across different time periods.
- In non-rush hours, the models demonstrate stable performance. However, during rush hours, there is a significant

performance gap, suggesting that the ST-LSTM model can better capture ridership variations. This finding highlights the robustness of the ST-LSTM model in accurately predicting and adapting to changes in ridership patterns during high-demand periods.

Model interpretability

To address the limitation of LSTM models in capturing spatial information, we have introduced a spatial correlation learning module to extract spatial features between origins and destinations (ODs) within the entire network. To further investigate the impact and significance of this module, as well as the influence of different values for x (representing the number of ODs with higher spatial correlation to the OD to be predicted), we conducted experiments using the MetroNJ2018 dataset as an example. Specifically, we focused on predicting the OD between Liuzhou East Road Station and Nanjing Forestry University Xinzhuang Station (LER-NFU). The data in Table 11 were obtained using equations (17)–(23), where p_{ij} represents the trend influence on LER-NFU, v_{ij} denotes the normalized ridership contribution to LER-NFU, and g_{ij} signifies the normalized location correlation. The weights of p_{ij} , v_{ij} , and g_{ij} for each OD were linearly calculated according to Eq. (23). Furthermore, by tuning the parameters, we set $\omega_1 = \omega_2 = 1$ and $\omega_3 = 0.8$, which allowed us to obtain a comprehensive ranking of the spatial correlation index z_{ij} . Finally, we identified the top 10 ODs with the highest correlation to LER-NFU, as presented in Table 11.

We conducted experiments by setting x from 0 to 10 to assess its impact on the model's performance. Specifically, when x is set to 0, the ST-LSTM model is equivalent to the TS-LSTM (No Spatial Correlation) model, where the

Table 9 The performance of prediction models under different temporal scenarios on MetroNJ2018

Models	Non-rush hours			Rush hours		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.098	4.138	0.2550	6.923	4.672	0.2785
ARIMA	5.870	3.944	0.2492	6.664	4.386	0.2708
RNN	4.585	3.177	0.1792	5.378	3.868	0.2289
NAR	4.310	3.018	0.1749	5.130	3.777	0.2168
LSTM	4.317	3.000	0.1775	5.103	3.755	0.2149
ConvLSTM	4.176	2.956	0.1747	4.993	3.697	0.2046
ST-LSTM (no temporal feature)	2.990	1.911	0.1008	3.854	2.669	0.1145
ST-LSTM (no spatial correlation)	2.946	1.893	0.0994	3.755	2.640	0.1131
ST-LSTM (no data inflow control)	2.930	1.871	0.0978	3.738	2.629	0.1135
ST-LSTM (no inflow)	2.873	1.840	0.0971	3.689	2.605	0.1126
ST-LSTM (no outflow)	2.825	1.818	0.0957	3.631	2.581	0.1101
ST-LSTM (ours)	2.787	1.666	0.0942	3.597	2.387	0.1076

Table 10 The performance of prediction models under different temporal scenarios on MetroBJ2019

Models	Non-rush hours			Rush hours		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE
SVR	6.426	4.471	0.2787	7.238	5.005	0.3021
ARIMA	6.214	4.272	0.2714	6.996	4.714	0.2930
RNN	4.938	3.489	0.2039	5.719	4.180	0.2535
NAR	4.526	3.362	0.2003	5.484	4.126	0.2288
LSTM	4.678	3.330	0.1998	5.452	4.085	0.2372
ConvLSTM	4.479	3.222	0.1973	5.319	4.025	0.2361
ST-LSTM (no temporal feature)	3.160	2.235	0.1287	4.011	2.993	0.1424
ST-LSTM (no spatial correlation)	3.119	2.240	0.1309	3.916	2.987	0.1446
ST-LSTM (no data inflow control)	3.089	2.199	0.1242	3.884	2.957	0.1399
ST-LSTM (no inflow)	3.047	2.168	0.1283	3.851	2.933	0.1438
ST-LSTM (no outflow)	3.014	2.156	0.1250	3.808	2.918	0.1394
ST-LSTM (ours)	2.955	1.994	0.1254	3.753	2.715	0.1388

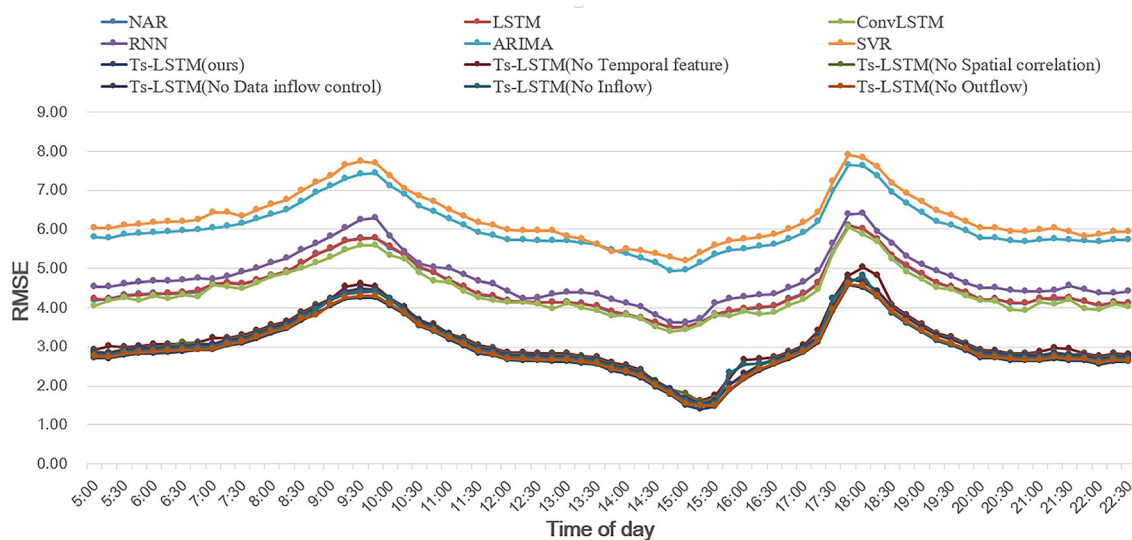


Fig. 8 The model performance comparison in different time intervals

Table 11 The top 10 ODs with the highest correlation with LER-NFU

Rank	OD Name	P_{ij}	v_{ij}	g_{ij}	z_{ij}
1	Tianrun City-Nanjing Forestry University	0.986	0.140	1.000	1.927
2	Liuzhou East Road-Wutang Square	0.973	0.285	0.692	1.811
3	Liuzhou East Road-Shangyuan Gate	0.985	0.220	0.681	1.750
4	Tianrun City-Wutang Square	0.988	0.137	0.632	1.631
5	Tianrun City-Shangyuan Gate	0.988	0.198	0.544	1.621
6	Tianrun City-Nanjing Station	0.982	0.168	0.547	1.588
7	Liuzhou East Road-Shanghai Road	0.975	0.124	0.415	1.431
8	Tianrun City-Xuanwu Gate	0.960	0.076	0.441	1.389
9	Liuzhou East Road-Bell Tower	0.953	0.170	0.297	1.361
10	Liuzhou East Road-Xian Gate	0.975	0.047	0.306	1.267

Table 12 The performance of prediction model with different inputs

Input x	Evaluation Indices			Improvements			Time requires (s)
	RMSE	MAE	MAPE	RMSE (%)	MAE (%)	MAPE (%)	
0	3.236	1.903	0.1071	0.00	0.00	0.00	21.618
1	3.138	1.858	0.1032	3.03	2.36	3.68	66.931
2	3.091	1.823	0.1016	4.50	4.22	5.18	113.218
3	3.070	1.812	0.1006	5.16	4.79	6.03	158.423
4	3.065	1.808	0.1003	5.29	5.00	6.35	204.821
5	3.061	1.804	0.1001	5.42	5.18	6.50	249.842
6	3.057	1.801	0.1000	5.55	5.36	6.65	297.099
7	3.049	1.798	0.0999	5.80	5.52	6.78	341.686
8	3.045	1.795	0.0997	5.93	5.67	6.91	384.887
9	3.041	1.792	0.0996	6.05	5.82	7.03	425.176
10	3.037	1.789	0.0995	6.15	5.97	7.12	471.383

spatial correlation learning module is removed. The results are presented in Table 12. From the table, we can observe that as x increases, the performance of the ST-LSTM model improves. This finding further confirms the effectiveness and significance of the proposed spatial correlation learning module. However, as x increases, the program's execution time increases significantly. Therefore, we must balance performance and computational efficiency and select a value that provides acceptable performance and reasonable execution time. We found that when x reaches 3, the performance improvement rate decreases significantly. Hence, for this paper, we chose $x = 3$ as the input value for the ST-LSTM model.

Conclusion

This study presents an enhanced spatial-temporal long short-term memory model (ST-LSTM) for conducting short-term OD prediction in URT (urban rail transit). The proposed model incorporates several innovative components, including a temporal feature extraction module, spatial correlation

learning module, and data inflow control gate. These components are specifically designed to address the unique challenges associated with URT OD prediction. Using two real-world datasets, we evaluated the model's performance across different spatial-temporal scenarios. The experimental results consistently demonstrate that the ST-LSTM model outperforms other baseline models, showcasing its powerful capability to learn and leverage spatial-temporal correlations effectively.

The main findings of the study can be summarized as follows: (1) Utilizing data that exhibit similar travel patterns can enhance the model's computational efficiency and prediction accuracy; (2) Extracting spatial correlation in large-dimensional and uneven OD flow is indeed a critical challenge in urban rail transit (URT) research. To tackle this problem, we propose a spatial correlation module that leverages multi-source data to pre-select highly correlated OD pairs from the entire network and integrates them into the model input. This innovative approach significantly enhances prediction accuracy. Moreover, this method also offers good interpretability; (3) Real-time OD flow data are often unavailable due to travel time. To address this issue, we introduce

a data inflow control gate that combines the available OD flow data with real-time inflow/outflow data. This integration significantly improves the model's performance by incorporating up-to-date information into the predictions; (4) The prediction time slot significantly affects the model's performance. A short time slot enhances the randomness of OD flow and reduces computational efficiency. However, a long time slot reduces the training data, potentially leading to poor model accuracy. Thus, it is essential to select an appropriate time slot that strikes a balance between computational efficiency and prediction accuracy.

However, this work does have some limitations. First, due to the limited availability of only one month's worth of data, our study focused on extracting temporal features at the daily level. However, with access to more extensive data, future research could investigate incorporating differences in OD flow during various time slots within a day to enhance prediction accuracy further. Second, we could only sample a subset of the ODs for evaluating our model due to computational resource constraints. As a result, there is still a gap between the proposed model and its real-world application. Moreover, since we lacked access to external data, our model does not take into account the influence of weather conditions, accidents, and other factors. In future research, we plan to enhance and optimize our model to ensure efficient application on a larger scale. Additionally, incorporating multi-source data such as weather information, festivals, and accident records can further enhance the performance of the proposed model.

Acknowledgements This work was supported by the Fundamental Research Funds for the Central Universities (Project No. 2022YJS067) and the National Natural Science Foundation of China (Project Nos. 72288101 and 72091513).

Data availability The Automatic Fare Collection (AFC) dataset used in this study is subject to a confidentiality agreement and cannot be shared due to privacy and legal restrictions. Access to the dataset is limited to authorized personnel within the organization that holds the data. Researchers interested in accessing the data should contact Beijing Urban Construction design & Development Group Co., Ltd to inquire about potential collaboration or data access procedures, which may be subject to further review and approval.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material

is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Bai L, Yao L, Wang X, Li C, Zhang X (2021) Deep spatial-temporal sequence modeling for multi-step passenger demand prediction. *Fut Gener Comput Syst* 121:25–34. <https://doi.org/10.1016/j.future.2021.03.003>
- Chai D, Wang L, Yang Q (2018) Bike flow prediction with multi-graph convolutional networks. In: Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems. ACM, SIGSPATIAL'18. <https://doi.org/10.1145/3274895.3274896>
- Chu KF, Lam AYS, Li VOK (2020) Deep multi-scale convolutional LSTM network for travel demand and origin-destination predictions. *IEEE Trans Intell Transp Syst* 21(8):3219–3232. <https://doi.org/10.1109/tits.2019.2924971>
- Feng S, Cai G (2016) Passenger flow forecast of metro station based on the ARIMA model. In: Proceedings of the 2015 international conference on electrical and information technologies for rail transportation. Springer, Berlin, pp 463–470. https://doi.org/10.1007/978-3-662-49370-0_49
- Guo S, Lin Y, Feng N, Song C, Wan H (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: National conference on artificial intelligence. <https://doi.org/10.48550/arXiv.2302.12973>
- Han Y, Wang S, Ren Y, Wang C, Gao P, Chen G (2019) Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. *ISPRS Int J Geoinform* 8(6):243. <https://doi.org/10.3390/ijgi8060243>
- Hochreiter S, Schmidhuber J (1997) Long short-term memory. *Neural Comput* 9(8):1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Hu W, Yan L, Liu K, Wang H (2015) A short-term traffic flow forecasting method based on the hybrid PSO-SVR. *Neural Process Lett* 43(1):155–172. <https://doi.org/10.1007/s11063-015-9409-6>
- Jaafar BA, Gaata MT, Jasim MN (2020) Home appliances recommendation system based on weather information using combined modified k -means and elbow algorithms. *Indones J Electr Eng Comput Sci* 19(3):1635. <https://doi.org/10.11591/ijeecs.v19.i3.pp1635-1642>
- Jiang X, Jia F, Feng J (2019) An online estimation method for passenger flow OD of urban rail transit network by using AFC data. *J Transp Syst Eng Inf Technol* 18(5):129–135
- Jin G, Wang M, Zhang J, Sha H, Huang J (2022) STGNN-TTE: travel time estimation via spatial-temporal graph neural network. *Fut Gener Comput Syst* 126:70–81. <https://doi.org/10.1016/j.future.2021.07.012>
- Ke J, Zheng H, Yang H, Chen XM (2017) Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach. *Transp Res Part C Emerg Technol* 85:591–608. <https://doi.org/10.1016/j.trc.2017.10.016>
- Ke J, Qin X, Yang H, Zheng Z, Zhu Z, Ye J (2021) Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. *Transp Res Part C Emerg Technol* 122:102858. <https://doi.org/10.1016/j.trc.2020.102858>
- Kingma D, Ba J (2014) Adam: a method for stochastic optimization. <https://doi.org/10.48550/arXiv.1412.6980>

15. Li D, Cao J, Li R, Wu L (2020) A spatio-temporal structured LSTM model for short-term prediction of origin-destination matrix in rail transit with multisource data. *IEEE Access* 8:84000–84019. <https://doi.org/10.1109/access.2020.2991982>
16. Li J, Li W, Lian G (2022) A nonlinear autoregressive model with exogenous variables for traffic flow forecasting in smaller urban regions. *Promet* 34(6):945–957. <https://doi.org/10.7307/ptt.v34i6.4145>
17. Li X, Pan G, Wu Z, Qi G, Li S, Zhang D, Zhang W, Wang Z (2012) Prediction of urban human mobility using large-scale taxi traces and its applications. *Front Comput Sci* 2012:1
18. Li Y, Lu J, Zhang L, Zhao Y (2017) Taxi booking mobile app order demand prediction based on short-term traffic forecasting. *Transp Res Rec J Transp Res Board* 2634(1):57–68. <https://doi.org/10.3141/2634-10>
19. Liu L, Chen RC (2017) A novel passenger flow prediction model using deep learning methods. *Transp Res Part C Emerg Technol* 84:74–91. <https://doi.org/10.1016/j.trc.2017.08.001>
20. Liu L, Qiu Z, Li G, Wang Q, Ouyang W, Lin L (2019) Contextualized spatial-temporal network for taxi origin-destination demand prediction. *IEEE Trans Intell Transp Syst* 20(10):3875–3887. <https://doi.org/10.1109/tits.2019.2915525>
21. Liu Y, Zheng H, Feng X, Chen Z (2017) Short-term traffic flow prediction with conv-LSTM. In: 2017 9th international conference on wireless communications and signal processing (WCSP). IEEE. <https://doi.org/10.1109/wcsp.2017.8171119>
22. Lu CC, Zhou X, Zhang K (2013) Dynamic origin-destination demand flow estimation under congested traffic conditions. *Transp Res Part C Emerg Technol* 34:16–37. <https://doi.org/10.1016/j.trc.2013.05.006>
23. Lv Y, Duan Y, Kang W, Li Z, Wang FY (2014) Traffic flow prediction with big data: a deep learning approach. *IEEE Trans Intell Transp Syst*. <https://doi.org/10.1109/tits.2014.2345663>
24. Ma X, Tao Z, Wang Y, Yu H, Wang Y (2015) Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transp Res Part C Emerg Technol* 54:187–197. <https://doi.org/10.1016/j.trc.2015.03.014>
25. Ma X, Dai Z, He Z, Ma J, Wang Y, Wang Y (2017) Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17(4):818. <https://doi.org/10.3390/s17040818>
26. Mahmoudabadi A, Ghazizadeh A (2013) A hybrid PSO-fuzzy model for determining the category of 85th speed. *J Optim* 2013:1–8. <https://doi.org/10.1155/2013/964262>
27. Martínez LM, Viegas JM (2013) A new approach to modelling distance-decay functions for accessibility assessment in transport studies. *J Transp Geogr* 26:87–96. <https://doi.org/10.1016/j.jtrangeo.2012.08.018>
28. Moreira-Matias L, Gama J, Ferreira M, Mendes-Moreira J, Damas L (2013) Predicting taxi-passenger demand using streaming data. *IEEE Trans Intell Transp Syst* 14(3):1393–1402. <https://doi.org/10.1109/tits.2013.2262376>
29. Shi X, Chen Z, Wang H, Yeung DY, Wong WK, Woo WC (2015) Convolutional LSTM network: a machine learning approach for precipitation nowcasting. *Adv Neural Inform Process Syst* 2015:802–810
30. Su G, Si B, Zhi K, Zhao B, Zheng X (2023) Simulation-based method for the calculation of passenger flow distribution in an urban rail transit network under interruption. *Urban Rail Transit* 9(2):110–126. <https://doi.org/10.1007/s40864-023-00188-z>
31. Sun L, Dai X, Xu Y (2018) Short-term origin-destination based metro flow prediction with probabilistic model selection approach. *J Adv Transp*. <https://doi.org/10.1155/2018/5942763>
32. Tsai CW, Hsia CH, Yang SJ, Liu SJ, Fang ZY (2020) Optimizing hyperparameters of deep learning in predicting bus passengers based on simulated annealing. *Appl Soft Comput* 88:106068. <https://doi.org/10.1016/j.asoc.2020.106068>
33. Wang D, Cao W, Li J, Ye J (2017) DeepSD: supply-demand prediction for online car-hailing services using deep neural networks. In: 2017 IEEE 33rd international conference on data engineering (ICDE). IEEE. <https://doi.org/10.1109/icde.2017.83>
34. Wang H, Zhang Q, Wu J, Pan S, Chen Y (2019) Time series feature learning with labeled and unlabeled data. *Pattern Recogn* 89:55–66. <https://doi.org/10.1016/j.patcog.2018.12.026>
35. Wang K, Ma C, Qiao Y, Lu X, Hao W, Dong S (2021) A hybrid deep learning model with 1DCNN-LSTM-attention networks for short-term traffic flow prediction. *Phys A* 583:126293. <https://doi.org/10.1016/j.physa.2021.126293>
36. Wang S, Cao J, Chen H, Peng H, Huang Z (2020) SeqST-GAN. *ACM Trans Spatial Algorithms Syst* 6(4):1–24. <https://doi.org/10.1145/3378889>
37. Wang Y, Yin H, Chen H, Wo T, Xu J, Zheng K (2019) Origin-destination matrix prediction via graph convolution: a new perspective of passenger demand modeling
38. Wilson A (1967) A statistical theory of spatial distribution models. *Transp Res* 1(3):253–269. [https://doi.org/10.1016/0041-1647\(67\)90035-4](https://doi.org/10.1016/0041-1647(67)90035-4)
39. Xi J, Jia F, Feng J (2019) An online estimation method for passenger flow OD of urban rail transit network by using AFC data. *J Transp Syst Eng Inform Technol* 18(5):129–135. <https://doi.org/10.16097/j.cnki.1009-6744.2018.05.019>
40. Xiong X, Ozbay K, Jin L, Feng C (2020) Dynamic origin-destination matrix prediction with line graph neural networks and Kalman filter. *Transp Res Rec J Transp Res Board* 2674(8):491–503. <https://doi.org/10.1177/0361198120919399>
41. Xu X, Jin X, Xiao D, Ma C, Wong SC (2021) A hybrid autoregressive fractionally integrated moving average and nonlinear autoregressive neural network model for short-term traffic flow prediction. *J Intell Transp Syst* 27(1):1–18. <https://doi.org/10.1080/15472450.2021.1977639>
42. Yang D, Chen K, Yang M, Zhao X (2019) Urban rail transit passenger flow forecast based on LSTM with enhanced long-term features. *IET Intel Transp Syst* 13(10):1475–1482. <https://doi.org/10.1049/iet-its.2018.5511>
43. Yang H, Iida Y, Sasaki T (1991) An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts. *Transp Res Part B Methodol* 25(5):351–363. [https://doi.org/10.1016/0191-2615\(91\)90028-h](https://doi.org/10.1016/0191-2615(91)90028-h)
44. Yang X, Xue Q, Ding M, Wu J, Gao Z (2021) Short-term prediction of passenger volume for urban rail systems: a deep learning approach based on smart-card data. *Int J Prod Econ* 231:107920. <https://doi.org/10.1016/j.ijpe.2020.107920>
45. Yang X, Xue Q, Yang X, Yin H, Qu Y, Li X, Wu J (2021) A novel prediction model for the inbound passenger flow of urban rail transit. *Inf Sci* 566:347–363. <https://doi.org/10.1016/j.ins.2021.02.036>
46. Yao H, Wu F, Ke J, Tang X, Jia Y, Lu S, Gong P, Ye J, Li Z (2018) Deep multi-view spatial-temporal network for taxi demand prediction. In: Proceedings of the AAAI conference on artificial intelligence, vol. 32(1). <https://doi.org/10.1609/aaai.v32i1.11836>
47. Yao H, Tang X, Wei H, Zheng G, Li Z (2019) Revisiting spatial-temporal similarity: a deep learning framework for traffic prediction. *Proc AAAI Conf Artif Intell* 33(01):5668–5675. <https://doi.org/10.1609/aaai.v33i01.33015668>
48. Ye Q, Li Z, Wu J, Cheng L (2022) Urban traffic dynamic OD prediction based on multi-source data. *CICTP 2022*:844–854
49. Yu B, Yin H, Zhu Z (2018) Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the twenty-seventh international joint conference on artificial intelligence, international joint conferences on artificial intelligence organization. <https://doi.org/10.24963/ijcai.2018/505>

50. Zhang J, Yu Z, Qi D (2016) Deep spatio-temporal residual networks for citywide crowd flows prediction. *Proc AAAI Conf Artif Intell*. <https://doi.org/10.1609/aaai.v31i1.10735>
51. Zhang J, Shen D, Tu L, Zhang F, Xu C, Wang Y, Tian C, Li X, Huang B, Li Z (2017) A real-time passenger flow estimation and prediction method for urban bus transit systems. *IEEE Trans Intell Transp Syst* 18(11):3168–3178. <https://doi.org/10.1109/tits.2017.2686877>
52. Zhang J, Chen F, Wang Z, Liu H (2019) Short-term origin-destination forecasting in urban rail transit based on attraction degree. *IEEE Access* 7:133452–133462. <https://doi.org/10.1109/access.2019.2941177>
53. Zhang J, Che H, Chen F, Ma W, He Z (2021) Short-term origin-destination demand prediction in urban rail transit systems: a channel-wise attentive split-convolutional neural network method. *Transp Res Part C Emerg Technol* 124:102928. <https://doi.org/10.1016/j.trc.2020.102928>
54. Zhang J, Chen F, Cui Z, Guo Y, Zhu Y (2021) Deep learning architecture for short-term passenger flow forecasting in urban rail transit. *IEEE Trans Intell Transp Syst* 22(11):7004–7014. <https://doi.org/10.1109/tits.2020.3000761>
55. Zhang K, Jia N, Zheng L, Liu Z (2019) A novel generative adversarial network for estimation of trip travel time distribution with trajectory data. *Transp Res Part C Emerg Technol* 108:223–244. <https://doi.org/10.1016/j.trc.2019.09.019>
56. Zhao Z, Chen W, Wu X, Chen PCY, Liu J (2017) LSTM network: a deep learning approach for short-term traffic forecast. *IET Intel Transp Syst* 11(2):68–75. <https://doi.org/10.1049/iet-its.2016.0208>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.