Taylor & Francis
Taylor & Francis Group

Transportmetrica A: Transport Science

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/ttra21

Correlation-based feature selection and parallel spatiotemporal networks for efficient passenger flow forecasting in metro systems

Cong Xiu, Shuguang Zhan, Jinyi Pan, Qiyuan Peng, Zhiyuan Lin & S.C. Wong

To cite this article: Cong Xiu, Shuguang Zhan, Jinyi Pan, Qiyuan Peng, Zhiyuan Lin & S.C. Wong (04 Apr 2024): Correlation-based feature selection and parallel spatiotemporal networks for efficient passenger flow forecasting in metro systems, Transportmetrica A: Transport Science, DOI: 10.1080/23249935.2024.2335244

To link to this article: https://doi.org/10.1080/23249935.2024.2335244

Published online: 04 Apr 2024.

TRANSPORTMETRICA A: TRANSPORT SCIENCE
https://doi.org/10.1080/23249935.2024.2335244

Taylor & Francis
Taylor & Francis Group

OPEN ACCESS

Correlation-based feature selection and parallel spatiotemporal networks for efficient passenger flow forecasting in metro systems

Cong Xiu [a,e], Shuguang Zhan [b], Jinyi Pan[c], Qiyuan Peng[a,e], Zhiyuan Lin[c] and S.C. Wong[d]

[a]School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, People's Republic of China; [b]School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei, People's Republic of China; [c]Institute for Transport Studies, University of Leeds, Leeds, UK; [d]Department of Civil Engineering, The University of Hong Kong; [e]National United Engineering Laboratory of Integrated and Intelligent Transportation, Chengdu, People's Republic of China

ABSTRACT

This paper presents a novel framework for predicting metro passenger flow that is both interpretable and computationally efficient. The proposed method first uses a correlation-based spatiotemporal feature selection strategy (Cor-STFS) to identify the optimal input scheme for the prediction model, effectively reducing unnecessary interference. The framework then introduces a new multivariate passenger flow prediction architecture called STA-PTCN-BiGRU, which combines a spatiotemporal attention (STA) mechanism, parallel temporal convolutional networks (PTCN), and bidirectional gated recurrent units (BiGRU) to capture the dynamic internal patterns of passenger flow. By utilising parallel computing, this architecture significantly reduces resource consumption. The effectiveness of the proposed approach is evaluated using four datasets from the Shanghai Metro. Experimental results show that the new method outperforms baseline approaches in terms of root mean square error (RMSE), mean absolute error (MAE), and symmetric mean absolute percentage error (SMAPE), achieving average reductions of 9.98%, 8.08%, and 13.29% in these metrics, respectively.

ARTICLE HISTORY
Received 14 June 2023
Accepted 21 March 2024

KEYWORDS
Traffic forecasting; Deep learning; Spatiotemporal dependence; Feature extraction; Metro domain knowledge

1. Introduction

With the construction and widespread adoption of public transportation systems, modern citizens have grown accustomed to and regularly rely on metros due to their punctuality, cost-effectiveness, and environmental friendliness (Hao, Lee, and Zhao 2019; Liu, Liu, and Jia 2019). However, the increasing frequency of usage has led to capacity limitations during peak hours, resulting in congestion and inconvenience for passengers at stations (He, Zhao, and Tsui 2023; Shi et al. 2020). Both the supply side (operating company) and the demand

CONTACT Shuguang Zhan shuguangzhan@hfut.edu.cn School of Automotive and Transportation Engineering, Hefei University of Technology, Hefei, People's Republic of China

side (passengers) of the public transportation systems must address this congestion issue to improve overall passenger service levels (Li, Zheng, and Jia 2024; Shi et al. 2020).

Currently, intelligent transportation systems are widely deployed in smart cities to reduce traffic congestion of metro systems (Nagy and Simon 2018; Tedjopurnomo et al. 2020). As a crucial component of these systems, passenger flow forecasting plays a vital role in providing scientific data support for both the supply side and the demand side. By obtaining and analysing data on passengers' usage patterns within the metro network, the supply side can make adjustments to the timetable, operational plans, and even expand the existing infrastructure to enhance passenger service capabilities. Simultaneously, the demand side can leverage the data to optimise travel options and routes, thereby mitigating potential travel congestion. Therefore, accurate traffic forecasting, specifically passenger flow forecasting, has emerged as a prominent research focus in the field of intelligent transportation (Liu et al. 2022; Xiu, Sun, and Peng 2022a; Zeng and Tang 2023).

Traffic forecasting has been extensively investigated by scholars from various perspectives (Nagy and Simon 2018; Zhang et al. 2011). Initially, research primarily concentrated on machine learning and statistical methods. With the availability of vast amounts of traffic data from intelligent transportation systems, data-driven traffic forecasting techniques, particularly deep learning (DL), have emerged as the dominant approach s. This includes recurrent neural networks (RNN), convolutional neural networks (CNN), graph convolutional neural networks (GCN), as well as their variations and combinations. These deep learning-based methods are widely applied in forecasting tasks such as traffic flow (Belletti et al. 2018; Lv et al. 2021), traffic congestion (Guo et al. 2021; Yang 2013), traffic speed (Ahn, Ko, and Kim 2016; Yu et al. 2017), passenger flow (Liu et al. 2022; Xue et al. 2022; Zhang et al. 2020), and passenger demand (Ke et al. 2017; Tang et al. 2021), yielding remarkable advantages and superior performance.

However, there are three key issues that require more attention in the research on passenger flow forecasting using deep learning.

- First, the selection of appropriate and interpretable data as input to neural network-based models is a challenging task (Tang, Alelyani, and And 2014). Traditional methods often use raw data directly as input, leading to the inclusion of irrelevant information and compromising prediction accuracy. Moreover, for large-scale metro networks, including all available input data results in a significant computational burden. Furthermore, due to the black-box nature of neural networks, users are unable to conduct meaningful and interpretable analysis of the model's output based on the initial input data.
- Second, relying solely on a single model makes it difficult to achieve further improvements in accuracy (Ma et al. 2022). It is necessary to develop an efficient combination framework that can effectively handle the spatial-temporal characteristics of passenger flow (Lee and Rhee 2022). Existing research on combining models often suffers from long computation time and excessive memory consumption.
- Third, when considering global external features, current approaches often fail to incorporate domain knowledge specific to the object being predicted (Liu, Liu, and Jia 2019). Previous studies have focused solely on advanced forecasting models for metro passenger flow prediction, neglecting the unique operational properties of metros. For instance, trains adhere to precise timetables, and stations with similar characteristics may exhibit similar passenger flow distributions over time and space.

To address the challenges mentioned above, this paper presents a novel deep learning-based framework for predicting passenger flow in metro systems, drawing inspiration from the works of Guo et al. (2022) and Ma et al. (2022). First, a spatial-temporal feature selection algorithm (Cor-STFS) is introduced into the framework to extract highly correlated data in an interpretable manner. Specifically, this algorithm determines the best input framework for the model based on the Pearson correlation coefficient, which can significantly improve the model's performance. Second, a parallel computing-based model called STA-TCN-BiGRU is proposed for the prediction framework. Unlike traditional models that rely on RNNs for capturing contextual information, this model leverages temporal convolutional networks (TCN). TCNs have a longer memory and are capable of avoiding gradient problems encountered by RNNs (Bai, Kolter, and Koltun 2018). Moreover, TCNs support layer-wise computation, allowing for simultaneous weight updates at each time step and enabling parallel computation. To further enhance prediction accuracy, an attention mechanism is introduced to assign varying importance to different features. The attention mechanism requires fewer parameters than RNNs and supports efficient parallel computing without relying on previous space-time steps (Vaswani et al. 2017). Spatial and temporal attention (STA) are combined in a parallel computing manner to capture spatial-temporal dependencies. Additionally, bidirectional gated recurrent units (BiGRU) are incorporated to effectively capture bidirectional temporal dependencies, which are commonly used in combination models (Guo et al. 2022). Thus, the DL-based framework STA-PTCN-BiGRU is proposed for metro passenger flow prediction. Third, train timetable information is extracted to construct external features related to train activities, which are then integrated into the model using feature engineering and one-hot encoding. To reduce dimensionality, an embedding layer is employed due to the sparsity of these external features. Finally, the feasibility and effectiveness of the proposed method are validated using real data from four different types of stations in the Shanghai metro network in China. Importantly, our approach differs from those that solely rely on black-box neural network-based models by introducing two key modules to enhance interpretability. First, our method employs the Cor-STFR feature selection technique to identify specific spatiotemporal features that influence passenger flow predictions. Second, our model incorporates unique metro system attributes, such as train timetable data, as external features within the prediction framework, thereby integrating these crucial aspects. The main contributions can be summarised as follows:

- The Cor-STFS, a feature selection method based on maximum correlation information, is introduced to improve the quality of input data. Specifically, interference from irrelevant information is reduced by retaining the most relevant spatial-temporal information as input for prediction using the Cor-STFS.
- The network framework, STA-PTCN-BiGRU, is designed for metro passenger flow prediction. Notably, a parallel architecture is employed in the STA-PTCN-BiGRU, effectively reducing the computational burden and achieving favourable results in prediction accuracy.
- Train timetable features are developed as global external features for passenger flow prediction. In particular, the train timetable feature, which considers the characteristics of the metro system, is incorporated into the prediction framework, further enhancing prediction accuracy.

- Excellent generalisation properties have been demonstrated for different types of metro stations by our model. The feasibility and effectiveness of the proposed method are evaluated using real data from the Shanghai Metro, providing detailed insights into its performance.

The rest of this paper is organised as follows. In Section 2, relevant research on traffic forecasting is reviewed. Section 3 describes the statement of the metro passenger flow forecasting problem. Section 4 explains the proposed framework and related methods. Section 5 presents the data description and experimental settings. Section 6 provides some numerical results to verify the effectiveness of the proposed method. In Section 7, the paper is concluded, and future work is discussed.

2. Literature review

2.1. Traffic state prediction

Traffic state prediction methods can be broadly categorised into two types: model-based and data-driven methods (Tedjopurnomo et al. 2020). Earlier research primarily focused on the former, such as the cell transmission model (Wei, Cao, and Sun 2013) and store-and-forward model (Aboudolas, Papageorgiou, and Kosmatopoulos 2009). These methods perform well when there are consistent traffic state changes. However, they struggle to accurately describe complex real-world traffic states and fail when changes are irregular (Lv et al. 2014). Consequently, data-driven methods have gradually replaced model-based methods in recent years.

With the advancements in data acquisition technology and the availability of traffic-related big data, data-driven methods have gained popularity(Zhang et al. 2011). Unlike model-based methods, data-driven methods leverage statistical regularities and distributions in historical data to infer changes in traffic states (Jiang et al. 2022; Shahriari et al. 2020). Existing data-driven methods fall into two categories: parametric-based models and non-parametric models (Nagy and Simon 2018; Zhang et al. 2011). Parametric models employ regression functions to forecast traffic based on historical data, like the autoregressive integrated moving average (ARIMA) and its variants, e.g. the Kohonen ARIMA (Lee and Fambro 1999), ARIMAX (Williams 2001) and seasonal ARIMA (Williams and Hoel 2003). Although these models are user-friendly, their predictive performance is limited due to their inability to handle nonlinearity and non-stationarity in traffic data. Non-parametric models have emerged as a solution to this problem (Shi et al. 2020; Zhang and Liu 2009). Machine learning-based approaches are the most representative, e.g, K-nearest neighbour (KNN), support vector machine (SVM) and Kalman filter method. With sufficient historical data, KNN (Arroyo and Maté 2009; Yu et al. 2019), SVM (Castro-Neto et al. 2009; Wu, Ho, and Lee 2004) and Kalman filter models (Kumar 2017) can learn statistical patterns and achieve more accurate predictions.

Inspired by the success of DL in computer vision and natural language processing, researchers have shown a preference for DL-based models based on neural networks due to their strong learning and generalisation capabilities (Lv et al. 2014; Tedjopurnomo et al. 2020). RNN and its variants, e.g. long short-term memory (LSTM) and gated recurrent unit

(GRU), have been widely used for traffic prediction (Sheu, Lan, and Huang 2009), demonstrating excellent performance in capturing long-term time dependencies. For example, the LSTM network is utilised for traffic speed prediction (Ma et al. 2015), travel time prediction (Duan, Lv, and Wang 2016), traffic flow prediction (Fu, Zhang, and Li 2016) and passenger flow prediction (Xiu et al. 2022b). However, existing methods that solely focus on temporal dependence while neglecting spatial dependence still have scope for improvement in terms of prediction accuracy. To address this, researchers have gradually introduced combined CNN and LSTM models, where CNN captures spatial dependence and LSTM captures temporal dependence (Sattarzadeh et al. 2023). For example, LSTM and CNN are combined for taxi demand prediction, and their method could benefit from modelling both spatial and temporal relations (Yao et al. 2018). Similar models are designed for network speed prediction (Ma et al. 2017), traffic flow forecasting (Zhang et al., 2019) and travel time prediction (Guo et al. 2022). As traffic networks are non-Euclidean, CNN-based models alone fail to fully capture spatial-temporal dependencies (Ye et al. 2022). To address this, GCN was proposed to construct spatial relationships between nodes (Chen et al. 2023). Li et al. (2017) proposed a diffusion convolutional recurrent neural network (DCRNN) that combines GCN and GRU in a Seq2Seq framework to predict traffic flow. The similar approach proposed by Zhao et al. (2020), namely T-GCN, has attracted extensive attention.

Furthermore, since TCNs can manage sequences in a causal manner and avoid the leakage of future information, they are comparable with LSTM frameworks in terms of the robustness and accuracy in modelling sequence problems (Bai, Kolter, and Koltun 2018). TCNs gradually replace RNN and its variants and combine with GCN for capturing spatial-temporal feature. Wu et al. (2019) proposed the graph wavenet (GWN), which combines TCN and GCN to capture spatial-temporal dependencies for traffic prediction. Additionally, attention mechanisms have proven effective in building dependencies in sequences and have been incorporated into traffic prediction models. For example, the attention based spatial-temporal graph convolutional network (ASTGCN) and its improvement, spatial-temporal graph networks (ASTGNN), utilise spatial and temporal attention mechanism to enhance network-level traffic prediction accuracy (Guo et al. 2019; 2021). Other similar models, such as graph multi-attention network (GMAN) and multivariate timeseries-based graph neural network (MTGNN), have also been developed and obtained advanced effects. Zheng et al. (2020) designed the GMAN model that combines temporal and spatial attention to capture associations between traffic sensors. Wu et al. (2020) introduced the use of a graph learning layer in MTGNN to adaptively construct a graph structure for traffic flow prediction. Despite the encouraging results achieved in current research, there are still some limitations. While these deep learning methods have yielded excellent performance, little attention has been paid to improving the prediction performance from the view of input features. Some researchers aim to enhance the quality of model input by combining models through feature selection (Zhang et al., 2019; Kim et al. 2020; Ma et al. 2022). However, these studies lacked adequate selection of spatial-temporal analysis features and did not establish a close integration with deep learning models. Furthermore, existing frameworks that involve model stacking require significant computing resources and entail lengthy training times (Tedjopurnomo et al. 2020). In terms of external features, the existing studies typically consider external factors such as the weather (Yuan et al. 2011; Zhang, Zheng, and Qi 2017) and points of interest (Geng et al. 2019; Lin et al. 2019) as global features. The inherent properties of a traffic system are rarely considered owing to the lack of external data resources. In practice, a metro system possesses various unique operating characteristics that affect short-term passenger flow, such as the punctuality of trains.

Table 1 provides a summary of studies closely related to our work. Specifically, we compare existing works based on three aspects: predicted goals, model structures (including spatial dependency and temporal dependency), and special techniques (such as external features, feature selection, and parallel architecture). Note that the predicted goal of the studies listed in Table 1 focuses on road or rail transportation systems. Several key points can be inferred from the table.

To capture temporal and spatial dependencies, the majority of advanced research employs a combination of GCN-based, CNN-based, or RNN-based models, with the notable exception of the study by Kim et al. (2020). Furthermore, only a small number of papers (Guo et al. 2021; Hao, Lee, and Zhao 2019) incorporate attention mechanisms to enhance the capture of temporal and spatial dependencies. Unlike the existing research listed in Table 1, our model combines TCN, BiGRU, and attention mechanisms simultaneously, fully leveraging the unique strengths of each module. Regarding external features, prevalent studies utilise time-related factors (Guo et al. 2021; Kim et al. 2020; Wu et al. 2019; Yu et al. 2017) or consider weather conditions (Hao, Lee, and Zhao 2019) to improve prediction performance. However, few papers incorporate the distinctive properties of the metro system, such as train events, to construct external features. As for specially designed feature selection methods, existing studies demonstrate that appropriate feature selection can increase the upper limit of predictive models (Tang, Alelyani, and And 2014). However, only a few studies (Kim et al. 2020; Ma et al. 2022) consider specialised feature selection with the aim of obtaining a comprehensive range of relevant information while avoiding unwanted interference. In terms of computational efficiency, existing forecasting models in Table 1 typically rely on stacked architectures with temporal and spatial dependencies. However, these architectures necessitate extensive training time and computing resources, thus limiting their practical applicability.

To clearly understand the focus of this research, we differentiate our study from two closely related recent pieces by Guo et al. (2022) and Ma et al. (2022). Specifically, the primary distinctions between our approach and Guo et al. (2022) are as follows: our research targets passenger flow prediction in a metro system, while Guo et al. (2022) and Ma et al. (2022) focus on traffic state prediction, such as travel time and traffic speed in a road system. Notably, neither Guo et al. (2022) nor Ma et al. (2022) consider external features with specific properties of the transportation system, such as train events. Moreover, feature selection and parallel architecture are not considered in the work of Guo et al. (2022), while our approach incorporates both techniques. Lastly, we deviate from Guo et al. (2022) and Ma et al. (2022) in the choice of the prediction model. In addition to CNN and RNN-based networks, we design a novel parallel architecture based on temporal and spatial attention modules.

3. Problem statement

Considering the limitations of the existing models, this section describes the concept introduced in the model and constructs the designed forecasting form. It is proposed that

Table 1. Comparison with relevant studies.

Publication	Goal	Model	Spatial dependency	Temporal dependency	External features	FS	PA
Li et al. 2017	Traffic flow	DCRNN	GCN	GRU	Time	No	No
Wu et al. 2019	Traffic flow	GWN	GCN	TCN	Time	No	No
Hao, Lee, and Zhao 2019	Metro passenger flow	ASeq2Seq	Attention +LSTM-BiGRU	Attention LSTM-BiGRU	Time +Weather	No	No
Kim et al. 2020	Passenger demand	LR-LSTM	LSTM	LSTM	Time	Yes	No
Tang et al. 2021	Passenger demand	MC-STGCN	GCN	GRU	No	No	No
Guo et al. 2021	Traffic flow	ASTGNN	Attention+ TCN	Attention+ TCN	Time	No	No
Guo et al. 2022	Travel time	CNN-BiLSTM	CNN	BiLSTM	No	No	No
Ma et al. 2022	Traffic speed	STFSA-CNN-GRU	CNN-GRU	CNN-GRU	No	Yes	No
This study	Metro passenger flow	STA-PTCN-BiGRU	Attention+ TCN-BiGRU	Attention+ TCN-BiGRU	Time+Train Event	Yes	Yes

Note: **FS** denotes feature selection; **PA** denotes parallel architecture passenger flow prediction consists of three components: passenger state of the target station, short-term context-aware trend influenced by the network, and the combined impact of unexpected events.

Definitions:

(1) **Passenger state of target station** V_t: The goal of this paper is to forecast the ridership (inflow/outflow) of a specific station (target station) for a future time period based on historical data. Passenger state is a general concept that can refer to either inflow or outflow. The passenger state of the target station at time step *t* is denoted as V_t.

(2) **Short-term context-aware trend** R_t: The state of the target station is susceptible to the conditions of other stations within the network. Additionally, the short-term trends of the target station are influenced by different stations in the network during different time periods. Hence, incorporating the short-term trends resulting from network effects into the prediction task provides a significant advantage compared to solely relying on the target station's historical data. A metro network is defined as a directed graph. The node set of the network can be denoted as $S = \{S_1, S_2, .., S_n\}$, and each node represents a metro station. The short-term context-aware trend is obtained by the aggregation of the passenger flow characteristics for each station and is defined as $R_t = \{R_t^{(1)}, R_t^{(2)}, \ldots, R_t^{(n)}\}$ at time step *t*.

(3) **External events** E_t: External events at the target station, e.g. train activities, time-of-day and day-of-week, also influence the passenger state of target station. This paper incorporates station train activities as external features using metro domain knowledge. The external events of target station at time step *t* are denoted as E_t.

Thus, on the basis of passenger state of target station V_t, short-term context-aware trend R_t and external events E_t, the state features of the target station are defined as $X_t = \{V_t, R_t, E_t\}$. In this way, the multi-step forecasting problem can be considered as learning the key parameters of prediction model on the historical data and the observed state features, as shown in (1).

$$[X_{t-\tau+1}, X_{t-\tau+2}, \ldots, X_t] \xrightarrow{f} [V_{t+1}, V_{t+2}, \ldots, V_{t+L}] \tag{1}$$

where $f(.)$ is a mapping function aimed at learning, τ denotes the input length of the model, and *L* represents the output horizon.

4. Methodology

The overview of our proposed approach for metro passenger flow prediction is illustrated in Figure 1. As depicted, our approach begins with applying Cor-STFR to process raw passenger data from the metro AFC system. This initial step involves extracting passenger features with high correlation to initialise the input for the STA-PTCN-BiGRU prediction model. This prediction model consists of a parallel spatial-temporal attention module, a stacked TCN module, and a Bi-GRU module. Through parallel computing, the model captures the passenger-related spatiotemporal characteristics of the input sequence. Furthermore, the timetable feature is utilised to extract metro operation features using the one-hot encoding method, serving as external features. Finally, a feature fusion module is employed

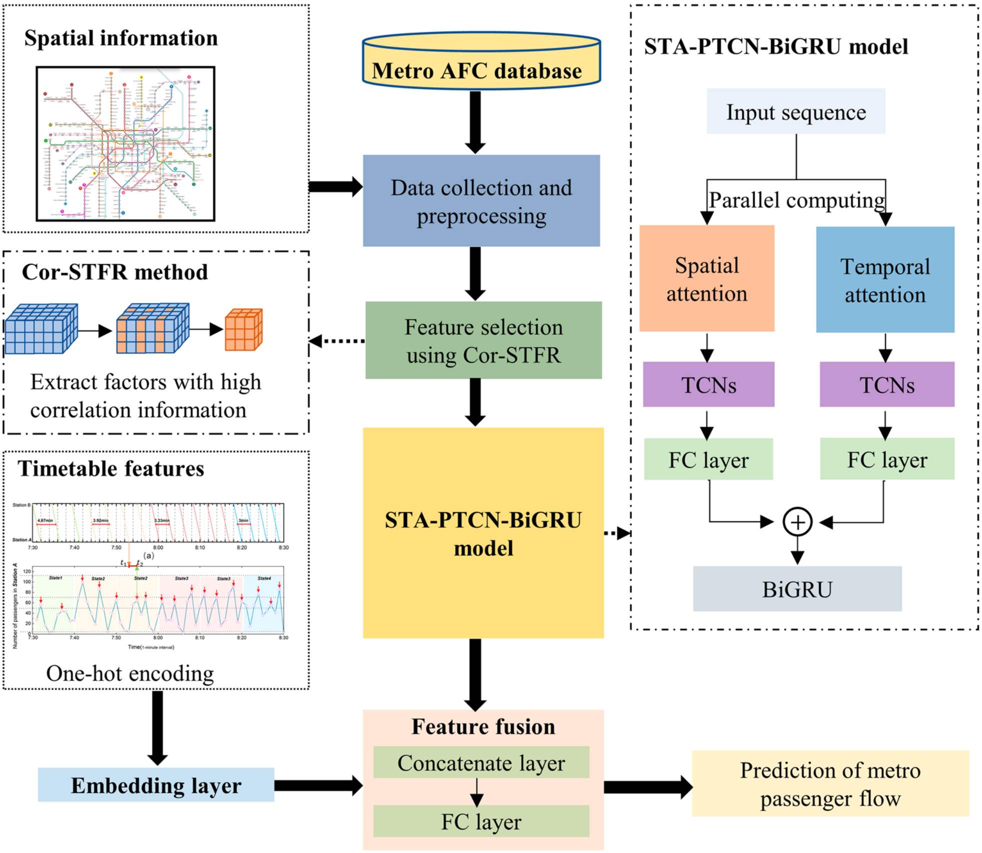

Figure 1. Overview of the proposed prediction approach.

to combine the spatiotemporal features with the external features, ultimately generating passenger flow prediction results.

4.1. Cor-STFS method

In real-world operation, the passenger flow state of a station is influenced by contextual trends within the network, which encompass both spatially and temporally correlated factors (Nagy and Simon 2018; Tedjopurnomo et al. 2020; Zhang et al. 2011). In the spatial dimension, the passenger flow state of the target station is dependent on the passenger flow status of other stations in the metro network, reflecting the spatial-temporal correlation of passenger flow between stations. In the temporal dimension, the current passenger flow state is often a continuation of previous states, indicating the temporal proximity and closeness of passenger flow state. Therefore, considering only the historical data of a single station would limit the performance of the predictive model. However, directly using raw spatial-temporal data from the metro network as prediction input has two drawbacks. First, it contains a significant amount of low-correlation noise that hinders the performance of the prediction model. Second, due to the large scale of real-world metro networks and the massive amount of historical data, directly feeding the data into the prediction model

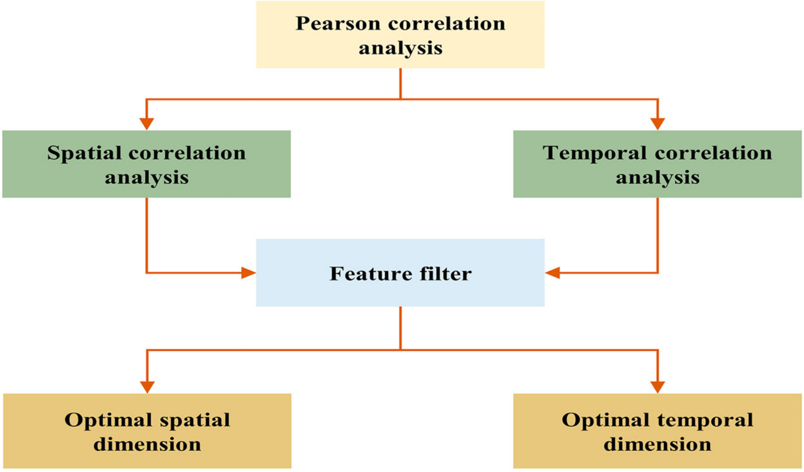
Pearson correlation analysis
Spatial correlation analysis
Temporal correlation analysis
Feature filter
Optimal spatial dimension
Optimal temporal dimension

Figure 2. The flowchart of Cor-STFS.

would rapidly consume computing resources. To address these challenges, a feature selection method called Cor-STFS is proposed, which is based on maximising correlations in the spatial-temporal domain. Cor-STFS serves as a preprocessing algorithm to identify input data that exhibits high predictive accuracy on the validation set.

As shown in Figure 2, the Cor-STFS algorithm consists of two main parts: spatial-temporal correlation analysis and determination of the optimal input. The correlation analysis assigns importance to input data by analysing the correlation coefficient between the passenger flow of each station and the target station. Through an iterative algorithm, the best input can be determined by considering both the temporal and spatial dimensions.

Spatial-temporal correlation analysis considering the context-aware trend feature enables the quantitative assessment of spatial and temporal factors. The context-aware trend feature of the metro network is denoted as $R_T = \{R_T^{(1)}, R_T^{(2)}, \ldots, R_T^{(j)}, \ldots, R_T^{(N)}\}$and is obtained by the aggregation of the passenger flow characteristics for each station within a specific period. Here, N represents the number of stations in the network, $R_T^{(j)} = [r_1^{(j)}, r_2^{(j)}, \ldots, r_T^{(j)}]$ denotes the passenger flow characteristics of the j-th station, and T denotes the number of time steps in the series feature. The historical passenger flow state of the target station is denoted as $V_T = [v_1, v_2, \ldots, v_T]$.

In this study, the Pearson correlation coefficient is employed as the criterion to measure the strength of the correlation, which is defined as follows:

$$\rho(V_T, R_T^{(j)}) = \frac{\sum_{t=1}^{T} (v_t - \bar{v})(r_t^{(j)} - \bar{r}^{(j)})}{\sqrt{\sum_{t=1}^{T} (v_t - \bar{v})^2} \sqrt{\sum_{t=1}^{T} (r_t^{(j)} - \bar{r}^{(j)})^2}} \tag{2}$$

where $v_t$and $r_t^{(j)}$represent the passenger flow value of the target station and the j-th station at time step t, respectively. $\bar{v}$ and $\bar{r}^{(j)}$ represent the mean of the passenger flow value of the target station and the j-th station across all time steps T, respectively. The Pearson coefficient allows for the assessment of the correlation between two time series, with values closer to 1 indicating a higher degree of correlation.

Since not all stations have strong spatial-temporal correlations with the target station, it is reasonable to set a correlation threshold. This allows us to extract information from stations in the network that have a relatively high impact on the target station. The filtered set of stations can be defined as follows:

$$J_\pi = \{j|\rho(V_T, R_T^{(j)}) \geq \pi\, , j = 1, 2, \ldots, N\} \tag{3}$$

where π represents the spatial association threshold, $J_\pi$represents the set of selected stations.

Further, based on the correlation analysis, the aim of spatial-temporal feature selection is to identify the input features $I*$with the smallest prediction error in the raw data. In this study, the feature selection scheme is based on the Las Vegas method (Ma et al. 2022), which can be formulated as follows:

$$I* = arg \min_I (Error(f(I), \hat{Y})),\ I \in R \tag{4}$$

where $Error(\cdot)$represents the error function (MAE is selected as the evaluation metrics in this study), $f(\cdot)$ represents the neural network for prediction task, $\hat{Y}$ represents the real value, and R represents the raw data.

The Cor-STFS algorithm follows these specific steps: First, the best selective parameters are initialised, i.e. time lag and spatial association threshold. Note that the number of spatial correlations can be obtained using formula (2). Additionally, the initial spatial-temporal characteristics can be determined based on the number of spatial correlations and time lag. The initial spatial-temporal features are then inputted into the trained neural network to obtain the initial error, which serves as the initial value of global minimum error. Next, an iterative search is performed. In each iteration, the search begins in the temporal dimension, incrementing the value of the current time lag by 1. Considering the change in time lag during association analysis, the current input features are updated, and the new prediction error is obtained. If the newly-generated error is not greater than the minimum error, the current best time lag and the global minimum error are simultaneously updated. The spatial association threshold is then gradually reduced by 0.05, updating the current input features based on changes in the number of spatial associations and obtaining the new error of the predictive model. If the new error is not greater than the global minimum error, the current best space threshold and the global minimum error value are updated. The iterative search continues until the termination condition is met. Once the condition is satisfied, the programme exits the loop and outputs the best data features. The pseudocode of the Cor-STFS algorithm is given in Algorithm 1.

Through Cor-STFS, the final input features for the trained prediction model are obtained, where the temporal length, i.e. time lag, is denoted as T, and the number of spatial correlations is denoted as n. Note that the temporal length and spatial number of the features at this stage are different from the unprocessed ones, although similar notation is used to represent them for simplicity. Overall, Cor-STFS allows for the reconstruction of the spatial-temporal matrix using the validation dataset. The matrix contains the maximum correlation information from the original data.

Algorithm 1: Cor-STFS algorithm.

```
Input:1. The original input data $R$; 2. The trained prediction model $f$; 3. Termination condition: maximum number of iterations $\Theta$and tolerance value $\varepsilon$
Output: The best input data $I^*$
Step 1: Initialization;
        Initialize the iteration number $\theta = 1$; Initialize the local time lag $T = T_0$and the local spatial association threshold $\pi = \pi_0$; Initialize the best time lag $T^* = T_0$and the best spatial association threshold $\pi^* = \pi_0$; Initialize the local input feature $I = R(T_0, \pi_0)$and the best input feature $I^* = R(T_0, \pi_0)$; Initialize the global minimum error $E^* = Error(f(I), \hat{Y})$and the error record $E^0 = Error(f(I), \hat{Y})$
Step 2: Search for input that shows the lowest prediction error;
    Step 2.1: Search in temporal dimension
        Increase current time lag $T = T + 1$, and then update current input feature $I = R(T, \pi^*)$
        Calculate current error in temporal dimension $E_T = Error(f(I), \hat{Y})$
        If Current error is not larger than the least error $E_T \leq E^*$
          Update the best time lag $T^* = T$ and the minimum error $E^* = E_T$
        End If
    Step 2.2: Search in spatial dimension
        Decrease current spatial association threshold $\pi = \pi - 0.05$, and then update current input feature $I = R(T^*, \pi)$
        Calculate current error in spatial dimension $E_S = Error(f(I), \hat{Y})$
        If Current error is not larger than the minimum error $E_S \leq E^*$
                Update the best spatial association threshold $\pi^* = \pi$ and the least error $E^* = E_S$
        End If
Step 3: Record the best input data $I^* = R(T^*, \pi^*)$and the global minimum error in this iteration $E^\theta = E^*$;
Step 4: Termination condition
If $\theta \geq \Theta$ or $E^{\theta-1}$ - $E^\theta \leq \varepsilon$, then terminate the algorithm and output the best input data $I^*$; Otherwise, $\theta = \theta + 1$, go to Step 2.
```

4.2. STA-PTCN-BiGRU architecture

The STA-PTCN-BiGRU architecture comprises parallel computations of temporal attention and spatial attention module, as well as stacked TCN module and BiGRU module. The DL-based architecture feeds a multivariate spatiotemporal sequence into two parallel backbones. One backbone employs spatial attention module to capture the spatial correlation between network state and the target station state. The other backbone utilises temporal attention module to extract the temporal correlation among all time steps within the prediction window, i.e. temporal length. The outputs of the two attention blocks are then forwarded to two parallel and stacked TCN layers with identical structures. Following dilated convolution and residual connection operations of TCN, the results are passed and aggregated into Bi-GRU layers.

For simplicity, it is assumed that the input of each module, denoted as I_t, represents passenger flow data or its features at time step t. Meanwhile, $\tilde{I}_t$represents the output of each module at time step t.

4.2.1. Spatial-Temporal attention module

An attention mechanism can alleviate the complexity of the neural network model (Bahdanau, Cho, and Bengio 2014). Not all the input information needs to be fed to the network, as only certain task-related information is required to be selected as the input for the neural network. Figure 3 and Figure 4 illustrate the inter-layer transformation details of the temporal attention module and the spatial attention module, respectively.

As shown in Figure 3, for the spatial attention module, the input is denoted as $I_t = [I_t^{(1)}, I_t^{(2)}, \ldots, I_t^{(n)}]$, where n represents the number of spatial correlations, and t represents the time step within the current time window. First, a spatial attention weight vector p_t,

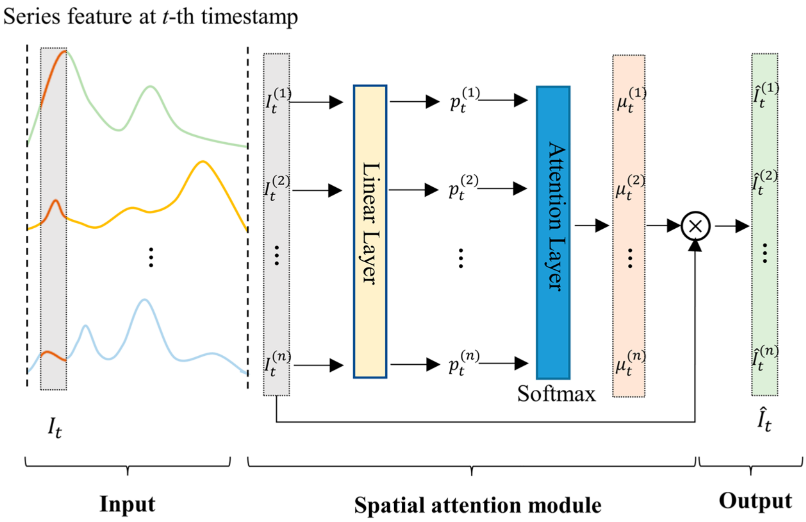

Figure 3. Data processing details in spatial attention module.

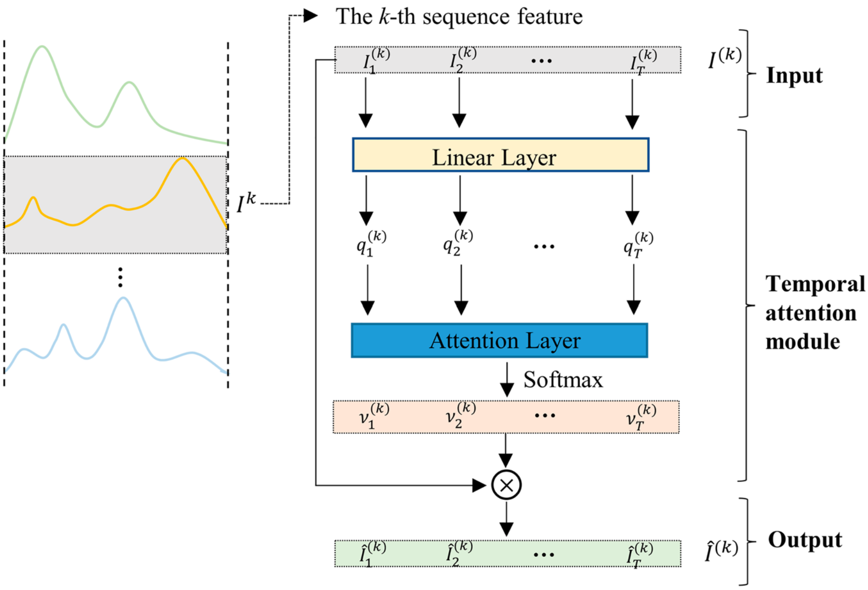
The k-th sequence feature
I^k
$I_1^{(k)}$ $I_2^{(k)}$ … $I_T^{(k)}$
$I^{(k)}$
Input
Linear Layer
$q_1^{(k)}$ $q_2^{(k)}$ … $q_T^{(k)}$
Attention Layer
Softmax
$v_1^{(k)}$ $v_2^{(k)}$ … $v_T^{(k)}$
Temporal attention module
$\hat{I}_1^{(k)}$ $\hat{I}_2^{(k)}$ … $\hat{I}_T^{(k)}$
$\hat{I}^{(k)}$
Output

Figure 4. Data processing details in temporal attention module.

which signifies the importance of each feature at time step t, is obtained through a linear transformation of the initial input I_t as follows:

$$p_t = W_p * I_t + b_p \tag{5}$$

where $W_p$and $b_p$are parameters that require learning. Then, the spatial weight p_t is normalised using the softmax function to ensure that the sum of all attention values is 1. Given the feature sequence index k, a regularised weight vector $\mu_t^{(k)}$ can be generated as follows:

$$\mu_t^{(k)} = \frac{\exp(p_t^{(k)})}{\sum\limits_k \exp(p_t^{(k)})} \tag{6}$$

The output of the spatial attention module is obtained by weighting the initial input with the normalised spatial attention vector $\mu_t^{(k)}$, as calculated by the following equation:

$$\tilde{I}_t = [\mu_t^{(1)} \cdot I_t^{(1)}, \mu_t^{(2)} \cdot I_t^{(2)}, \ldots, \mu_t^{(n)} \cdot I_t^{(n)}] \tag{7}$$

As shown in Figure 4, the temporal attention module takes the form $I^{(k)} = [I_1^{(k)}, I_2^{(k)}, \ldots, I_T^{(k)}]$ as input, where k represents the k-th sequence, and T represents the temporal length. Similar to spatial attention module, a linear transformation is applied to the input to generate the temporal attention weight vector $q^{(k)}$, representing the importance of the k-th sequence at all time steps within the time window:

$$q^{(k)} = W_q * I^{(k)} + b_q \tag{8}$$

The normalised temporal attention vectors are then processed through the softmax function:

$$\upsilon_t^{(k)} = \frac{\exp(q_t^{(k)})}{\sum\limits_t \exp(q_t^{(k)})} \tag{9}$$

The output of the time attention module is obtained by weighting the initial input with the normalised time attention vector $\upsilon_t^{(k)}$, calculated as follows:

$$\tilde{I}_t = [\upsilon_t^{(1)} \cdot I_t^{(1)}, \upsilon_t^{(2)} \cdot I_t^{(2)}, \ldots, \upsilon_t^{(n)} \cdot I_t^{(n)}] \tag{10}$$

4.2.2. Stacked TCN module

TCN, as a novel sequence modelling method, leverages the advantages of CNNs, offering increased parallelism and flexible receptive fields (Bai, Kolter, and Koltun 2018). The main components of TCN are casual dilated convolution and residual connection.

Casual dilated convolution is the combination of casual convolution and dilated convolution. The casual convolution ensures that there is no information leakage. Only the state at or before time step t is considered in casual convolution to calculate the output at time step t, which means that the feature extraction process excludes any future information. Dilated convolution is employed to address the challenge of processing exponentially growing long sequence data while preventing the network from becoming excessively deep. By using larger receptive fields, the network can achieve comparable performance with fewer training parameters and layers, which proves advantageous during training. The effective history length in casual dilated convolution is determined by $(K-1) \cdot d$, where K represents the kernel size and d is the dilated factor. To control the number of parameters, a fixed value for K is chosen and the value of d is exponentially increased layer by layer. Specifically, $d = 2^l$, where l denotes the level of the network.

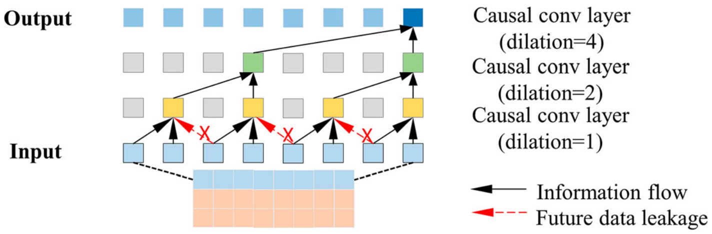
Output
Input
Causal conv layer (dilation=4)
Causal conv layer (dilation=2)
Causal conv layer (dilation=1)
Information flow
Future data leakage

Figure 5. Illustration of causal dilated convolutional structure.

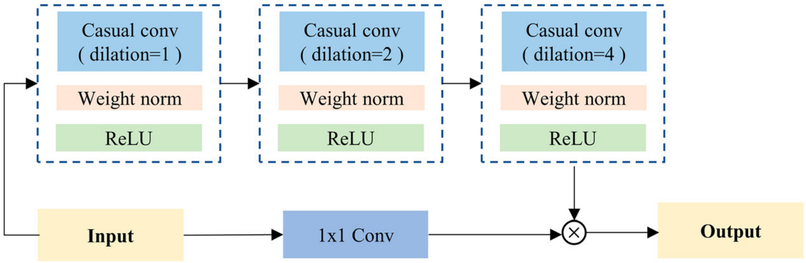
Casual conv (dilation=1)
Weight norm
ReLU
Casual conv (dilation=2)
Weight norm
ReLU
Casual conv (dilation=4)
Weight norm
ReLU
Input
1x1 Conv
Output

Figure 6. Illustration of residual connection structure.

Figure 5 illustrates interval sampling of causal dilated convolution. In comparison to traditional convolution methods that involve multilayer convolution and pooling, interval sampling effectively reduces information loss while preserving the same number of input and output time steps. Given the filter $F = \{f_0, f_1, \ldots, f_k, \ldots, f_{K-1}\}$, the causal dilated convolution process can be described as follows:

$$g_{h,t} = \sum_{k=0}^{K\text{ - }1} f_k \cdot g_{h-1,t-k\cdot d} \tag{11}$$

where $g_{h,t}$ represents the series value of the h-th layer in the network at time step t, K denotes the size of the convolution kernel, d signifies the convolution dilation factor, and $t - k \cdot d$ accounts for the direction of the past.

However, causal dilated convolution alone may not suffice when dealing with extremely long sequences, as it requires a deeper structure. Nonetheless, deep models can encounter the issue of gradient disappearance or explosion. To address this problem, residual connections are employed. The structure of residual connections is depicted in the Figure 6. TCN incorporates three layers of causal dilated convolution with dilation rates set to 1, 2, and 4, respectively. A unit convolution kernel of size 1 × 1 is utilised to process the original input, ensuring that the sequence structure remains consistent across the two paths during the summation operation. Residual connections enable the network to effectively propagate cross-layer information.

The residual connection is defined as summing input information and processing information through the TCN module as follows:

$$\tilde{I}_t = ReLU(I_t + G(I_t)) \tag{12}$$

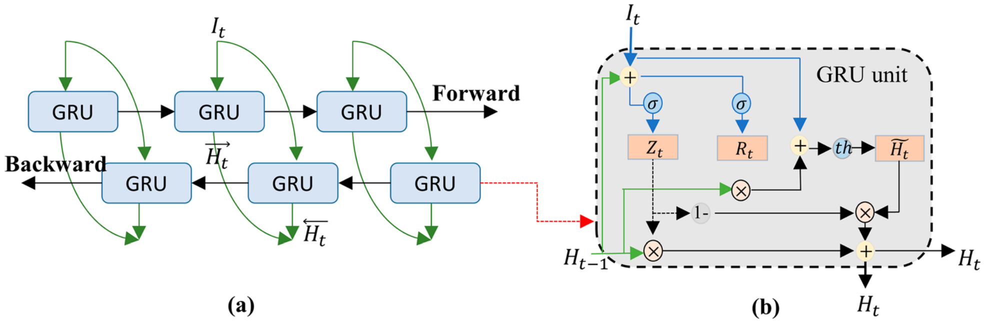

Figure 7. Illustration of BiGRU structure(a) and GRU structure(b).

where I_t represents the input of the TCN module, and $G(I_t)$represents the processing result of the causal dilated convolution.

4.2.3. BiGRU module

As an alternative to LSTM, the GRU network, with gate mechanisms e.g. reset gate and update gate and recurrent structures, learns relatively long-term dependencies (Cho et al. 2014). Moreover, the BiGRU is bidirectional, enabling the input sequences to be processed in both the forward and backward directions, thereby obtaining more comprehensive feature information. The structures of BiGRU and GRU are shown in Figure 7.

Specifically, reset gate $R_t = \{R_t^{(1)}, R_t^{(1)}, \ldots, R_t^{(n)}\}$, update gate $Z_t = \{Z_t^{(1)}, Z_t^{(1)}, \ldots, Z_t^{(n)}\}$, new information $\tilde{H}_t = \{\tilde{H}_t^{(1)}, \tilde{H}_t^{(1)}, \ldots, \tilde{H}_t^{(n)}\}$ and hidden state $H_t = \{H_t^{(1)}, H_t^{(1)}, \ldots, H_t^{(n)}\}$ can be calculated as follows:

$$\begin{cases} R_t = \sigma(W_{rx} * I_t + W_{rh} * H_{t-1} + b_r) \\ Z_t = \sigma(W_{zx} * I_t + W_{zh} * H_{t-1} + b_z) \\ \tilde{H}_t = tanh(W_{nx} * I_t + R_t \odot (W_{nh} * H_{t-1} + b_h)) \\ H_t = (1 - Z_t) \odot \tilde{H}_t + Z_t \odot H_{t-1} \end{cases} \tag{13}$$

where $\sigma$and $tanh$are activation functions, H_{t-1}represents the hidden state of the last iteration t-1. W_{rx}denotes the parameters between hidden state and input, and W_{rh} represent the parameters between R_t and H_{t-1}. Other parameters W_{zx}, W_{zh}, W_{nx} and W_{nh}share similar function that need to be learned. b_r, b_z and b_h represent bias terms. $\odot$ denotes the Hadamard Product, that is, multiplying the corresponding elements in the operation matrix.

Then, the hidden state resulting from the combined forward and backward calculations is fed into a fully connected layer, yielding the output of the BiGRU. This output can be computed using the following formula:

$$\tilde{I}_t = FC(\oplus(\overrightarrow{H_t}, \overleftarrow{H_t})) \tag{14}$$

where $\oplus(\cdot)$represents the concatenate operation, $\overrightarrow{H_t}$ and $\overleftarrow{H_t}$represent the forward and the backward hidden state, respectively. FCrepresents the fully connected layer.

4.3. External feature fusion

In previous studies, the usefulness of external features for traffic prediction has been demonstrated (Liu et al. 2022; Tedjopurnomo et al. 2020). However, existing research often overlooks the inherent characteristics of the transportation system (Liu, Liu, and Jia 2019; Xiu, Sun, and Peng 2022a). When predicting metro passenger flow, it is crucial to consider the influence of accurate timetables on passenger flow dynamics. With the availability of real-time timetable data, it is possible to extract this data to construct the disturbance characteristics caused by train activities on passenger flow. Therefore, in this study, the external features encompass both the train timetable features designed and the conventional time-of-day and day-of-week factors that have proven to be effective. The following section provides a detailed explanation of the construction of the train activity disturbance feature.

The operation of the metro possesses a unique feature, as trains strictly adhere to precise timetables. Consequently, passenger movement into and out of the station is constrained by the interference caused by train activity. To examine the influence of the metro timetable data on the passenger flow prediction, an analysis of the correlation between the passenger flow data and metro timetable data is conducted. The train arrival activity and outbound passenger flow for station A are used for data exploration. Figure 8(a) shows the timetable data between 7:30 and 8:30 am, obtained through the train schedule, in which different states characterise the variation in the train arrival interval. Figure 8(b) shows the fluctuations in the outbound passenger flow in the same period. Figure 8 shows a local peak in the alighting passenger flow when the train arrives. The peak occurs more frequently as the train's arrival interval decreases. For instance, as shown in Figure 8(a), two trains arrive at station A between 7:30 and 7:40 am (state 1), and three trains arrive in the 7:40–7:50 am period (state 2). Figure 8(b) shows that in the same period, the outbound flow of the same station (aggregated in 1 min intervals) has two peaks in states 1 and 3 in state 2. This correspondence indicates that train arrival events affect the appearance of passenger flow peaks. However, the time at which the train arrives (t_1) and that at which the peak occurs (t_2) do not coincide. This lack of coincidence is attributable to the fact that the peak passenger flow associated with the train occurs later than the arrival time of the corresponding train owing to the different walking speeds of various passenger groups and variation in the platforms' crowding conditions at different times of the day. Therefore, feature modelling is performed to characterise the correlation between the train arrival activities and passenger flow peaks. The feature is constructed considering two aspects: the location and intensity of train arrivals.

The first aspect is the temporal location characteristics of train arrivals, defined as the following binary vector:

$$U_t = [u_{t,1}, u_{t,2}, \ldots, u_{t,q}, \ldots, u_{t,d}] \tag{15}$$

$$u_{t,q} = \begin{cases} 1, & atrainarrivesatq - thpositionoft - thtimeinterval \\ 0, & otherwise \end{cases} \tag{16}$$

For example, given a time interval of 10 min and discrete time unit of 1 min (rounded up), the temporal position of the two trains (marked as green in Figure 4) that arrive at 7:32:00 and 7:36:40 is denoted as $[0, 1, 0, 0, 0, 0, 1, 0, 0, 0]$. In this vector, the 2nd and 7th positions equal 1 and other positions are set as zero.

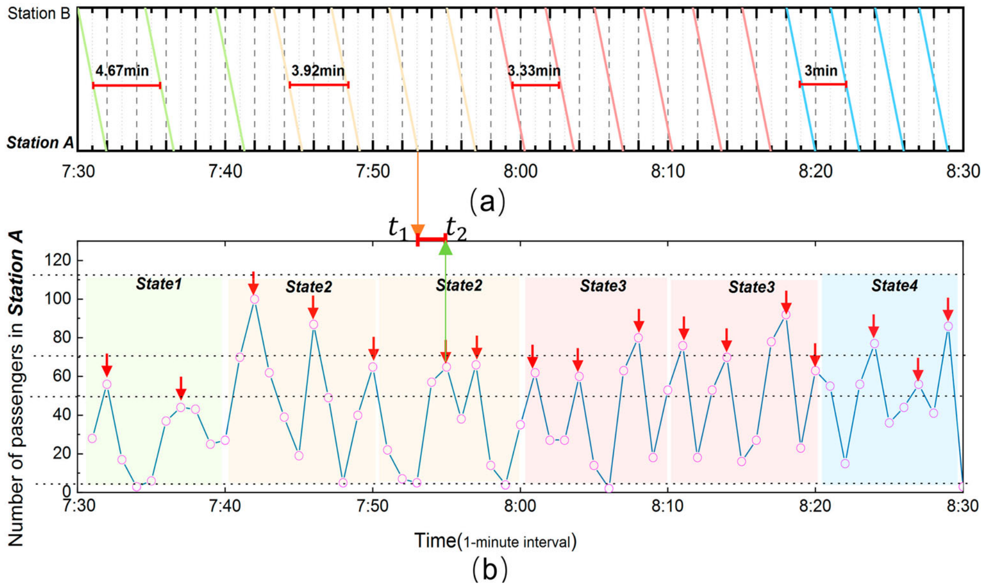

Figure 8. Correlation between (a) passenger flow data and (b) metro timetable data (1-min interval, 7:30–8:30; a state denotes a time interval of 10 min). Stations A and B are two adjacent stations on a metro network.

The second aspect is the intensity of train arrival, expressed as.

$$P_t = \sum_{q=1}^{d} u_{t,q} \tag{17}$$

where $P_t$is the total number of train arrivals in the t-th time interval.

The train temporal location and intensity characteristics are merged to establish the final train timetable features $M_t = [U_t; P_t]$ for a target station, where operator $[;]$ represents concatenation operator.

Regarding the time-of-day and day-of-week factors, specific rules are followed for classifying these factors based on previous studies (Liu, Liu, and Jia 2019; Xiu, Sun, and Peng 2022a), considering the timing and duration of passenger peaks throughout the day. The classification method is presented in Appendix A. One-hot encoding is utilised for the other external features (day-of-week and time-of-day), denoted as $T_t = [T_t^1; T_t^2]$, where $T_t^1 = onehot(\widetilde{T_t^1})$, $T_t^2 = onehot(\widetilde{T_t^2})$. $\widetilde{T_t^1}$and $\widetilde{T_t^2}$ represent the original day-of-week and time-of-day variables, while T_t^1 and T_t^2 represent the corresponding encoded external factor vectors.

The train timetable features and time-related external factors are then combined into a heterogeneous information matrix $E_t = [M_t; T_t]$, which enriches the multi-source features and serves as input for an embedding layer. It is important to note that, due to the high sparsity of the external factors, this study employs the embedding layer to transform the external factor matrix $E_t$into a vector of the same dimension as the preliminary prediction result.

As for the fusion module of the proposed method, the embedded external factor vector and the preliminary prediction result vector are concatenated to obtain the intermediate result $f_t = [\tilde{I}_t, E_t]$. Finally, the intermediate results are fed into a fully connected layer neural network to generate the final prediction results for metro passenger flow.

5. Experiments

5.1. Data description

5.1.1. Data source

The case study focuses on the Shanghai Metro, which is one of the largest urban rail transit systems globally, serving Shanghai, China, and the surrounding metropolitan area. With daily peak passenger numbers exceeding tens of millions, the Shanghai Metro provides an ideal case to validate the proposed method. The dataset used for this research was carefully compiled from verifiable, real-world data obtained from the Shanghai Metro system. This comprehensive dataset contains billions of transaction records collected from a network of 415 individual stations across 17 lines. The data collection period for this study spanned three months, from July 1 to September 30, 2019. Each transaction record in this dataset includes specific elements such as the transaction ID, entry and exit times, as well as the names of the corresponding entry and exit stations. The focus of the investigation is strategically centered on the time bracket of 6:00–23:00, as it has a significant impact on passenger forecasting. This operational focus aligns with previous relevant research in the field.

Table 2 shows a detailed overview of the experimental dataset. The dataset is organised by dividing the daily data into 102 distinct intervals, with each interval representing a duration of 10 min. The passenger data was sourced from an automatic fare collection system (AFC), while the train schedule data was obtained from an automatic vehicle location system (AVL). These extracted datasets were then combined to create comprehensive passenger flow and timetable datasets. The passenger flow dataset has a temporal dimension of 9384, covering 92 days with 102 observations per day. The spatial dimension of the dataset corresponds to the 415 stations in the metro system. The feature dimension of the passenger flow dataset consists of two features: inbound and outbound passenger flows. On the other hand, the timetable dataset includes two features: arrival temporal position and intensity. The experimental dataset was divided as follows: data from July 1 to September 2, 2019 (64 days) are used as the training set, data from September 3 to September 16 (14 days) are allocated as a validation set for model selection during the training process, and the remaining data from September 17 to September 30 (14 days) are used as the test set.

Further, to evaluate the effectiveness of the proposed method, four sample stations (Station S1, S2, S3, and S4) were selected, representing various passenger patterns within the metro network, as shown in Figure 9. Station S1 and Station S3 are transfer type, while Station S2 and Station S4 are non-transfer type. The inclusion of different station types allows for testing the generalisation and generality of our method.

5.1.2. Data spatial-temporal analysis

The spatiotemporal associations are also analysed using the validation dataset. The spatial distribution of impact levels for different target stations is shown in Figure 10. The

Table 2. Experimental dataset based on the Shanghai metro network in 2019.

Parameter	Value
Number of stations in the network	415
Time period for daily record	6:00 am–23:00 pm
Number of days in record	92 days
Tested time interval	10 min/ time step
Shape of overall passenger dataset	(9384,415,2)
Shape of timetable features	(9384,10)
Time period of training set	2019/7/1–2019/9/2 (64 days)
Time period of validation set	2019/9/3–2019/9/16 (14 days)
Time period of test set	2019/9/17–2019/9/30 (14 days)

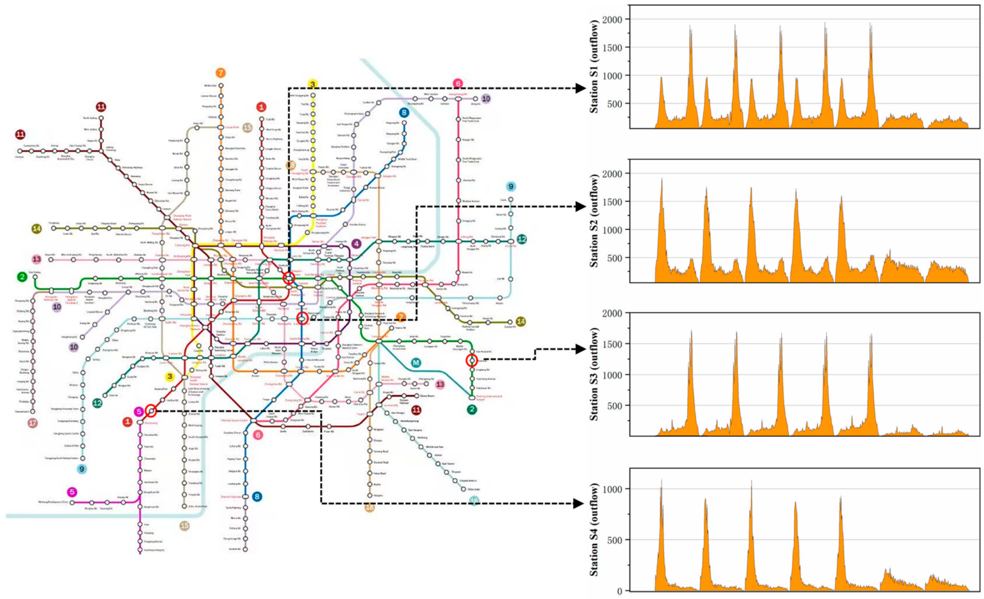

Figure 9. Distribution of tested stations in Shanghai metro system and the passenger patterns.

correlation coefficient indicates the strength of the connection between the station and the target station. A correlation coefficient closer to 1 signifies a stronger association. The analysis reveals distinct spatial associations among different station types. Figure 10(a,c) illustrate the spatial relationships of transfer-type stations, while Figure 10(b,d) represent non-transfer stations. For transfer-type target stations, those in proximity to the metro's core area tend to exhibit higher correlation coefficients, whereas stations farther away display lower coefficients. Conversely, non-transfer stations located further from the metro's core area generally exhibit higher correlation coefficients, likely due to shared traffic patterns. Interestingly, unlike road transportation systems, the correlation coefficient between two metro stations does not exhibit a significant relationship with their distance. Moreover, regardless of station type, the number of highly correlated stations is much smaller than the total number of stations, emphasising the importance of conducting spatial correlation analysis on the original data to reduce input factors in the spatial dimension.

Additionally, the Pearson correlation coefficient between the current passenger flow at time step *t* and the previous 12 timesteps (with each step representing 10 min) is calculated.

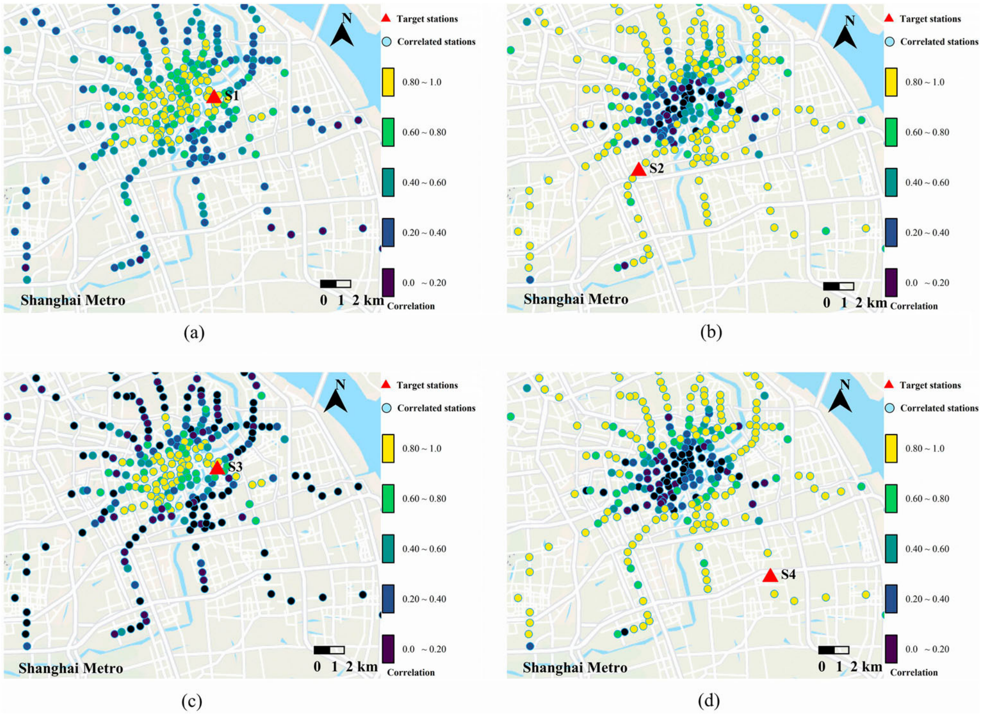

Figure 10. Spatiotemporal relationship results of the other stations in the network. (a) Station S1 (transfer type); (b) Station S2(non-transfer type); (c) Station S3(transfer type); (d) Station S4(non-transfer type).

Table 3. Pearson coefficients of passenger flow at different time intervals.

Time interval	Pearson coefficient	Time interval	Pearson coefficient
t-1 (0-10 min)	0.95	*t*-7 (60-70 min)	0.84
t-2 (10-20 min)	0.96	*t*-8 (70-80 min)	0.81
t-3 (20-30 min)	0.93	*t*-9 (80-90 min)	0.75
t-4 (30-40 min)	0.89	*t*-10 (90-100 min)	0.74
t-5 (40-50 min)	0.88	*t*-11 (100-110 min)	0.72
t-6 (50-60 min)	0.86	*t*-12 (110-120 min)	0.69

The results are presented in Table 3. It is observed that the correlation coefficient decreases as the analysis moves away from the current time period *t*, indicating stronger temporal correlations. However, in the *t*-9 period, the correlation coefficient is 0.75 (lower than 0.8), suggesting a relatively smaller influence of timing. Note that two random variables with a Pearson coefficient greater than 0.8 are generally considered highly correlated. Therefore, Therefore, the input factor of the temporal dimension can be reduced by selecting a reasonable and moderate time lag.

5.2. Experimental settings

5.2.1. Evaluation metrics

Three metrics are used to evaluate the prediction performance of the proposed method.

(1) Root mean square error (RMSE):

$$RMSE = \sqrt{\frac{1}{\Omega}\sum_{i=1}^{\Omega}(\hat{y}_i - y_i)^2} \tag{18}$$

(2) Mean absolute error (MAE):

$$MAE = \frac{1}{\Omega}\sum_{i=1}^{\Omega}|\hat{y}_i - y_i| \tag{19}$$

(3) Symmetric mean absolute percentage error (SMAPE):

$$SMAPE = \frac{1}{\Omega}\sum_{i=1}^{\Omega}\frac{|\hat{y}_i - y_i|}{(|\hat{y}_i| + |y_i|)/2} \times 100\% \tag{20}$$

where $\hat{y}_i$ and y_i are predicted value and actual value, respectively, at the i-th sample. Ω denotes the number of samples in the validation or test dataset.

5.2.2. Benchmark methods

Three classic machine learning models, four mainstream deep learning models, and three advanced graph-based learning models are selected as benchmark models. The following are brief introductions to their implementation details:

(1) Machine learning models:

- ARIMA: This traditional and widely used method combines autoregression with a moving average model for time series prediction. The autoregressive term, difference order, and moving average term are optimised based on the Akaike information criterion.
- KNN: K-Nearest Neighbours is a classic machine learning approach that calculates similarity using Euclidean distance and predicts by taking the mean value of neighbouring points. The number of neighbours is tuned from 1 to 14.
- MLP: Multi-layer perception is the most basic deep learning method. An MLP model is constructed, consisting of one input layer, two hidden layers, and one output layer.

(2) Deep learning models:

- LSTM-FC: This deep learning-based model utilises a long short-term memory network with fully connected hidden units to address the issue of vanishing gradients in standard RNNs. The hidden size of each LSTM layer is set to 256.
- GRU-FC: Similar to LSTM, this model employs gated recurrent units to capture sequential dependencies by replacing the LSTM layers. The hidden size of each GRU layer is set to 256.
- ASeq2Seq: A sequence-to-sequence model consisting of two fully-connected GRU layers with an attention mechanism for time series prediction. The hidden size of each GRU layer is set to 256.

- DARNN: This model incorporates dual-stage attention to capture dependencies in both input data and encoder hidden states.

(3) Graph-based learning:

- DCRNN (Li et al. 2017): A graph-based learning model that captures spatial dependencies using bidirectional random walks on graphs and learns temporal dependencies with an encoder-decoder architecture. This method is re-implemented based on the official code for metro ridership prediction.
- GWN (Wu et al. 2019): This method utilises an adaptive dependency matrix to capture hidden spatial dependencies and employs stacked dilated 1D convolution components to handle long sequences. This method is re-implemented based on the official code for metro passenger flow prediction.
- MTGNN (Wu et al. 2020): A model based on Graph Neural Networks (GNN) and Convolutional Neural Networks (CNN) that uses adaptive graphs, mix-hop propagation layers, and dilated inception layers to capture spatial-temporal correlations. This method is re-implemented based on the official code for metro passenger flow prediction.
- ASTGNN (S. Guo et al. 2021): An encoder-decoder architecture with residual connections and layer normalisation designed to learn the dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. This method is re-implemented based on the official code for metro passenger flow prediction.

5.2.3. Experimental settings

The proposed method utilises the MAE loss function and the adaptive moment estimation (Adam) optimiser. Historical data is fed into the proposed STA-PTCN-BiGRU network for training. The optimiser adjusts the parameters based on the training error. An initial learning rate of 0.001 is set. For the single-step prediction task, the prediction horizon is set at 10 min. In contrast, for multi-step prediction tasks, the horizon extends up to 120 min (12 steps). The number of look-back steps is consistently set to 8 (80 min before the prediction time) for all models. Specifically, in the single-step prediction task, we use data from the past 80 min to forecast passenger flow for the next 10 min. For multi-step predictions, we continue to employ the past 80 min of data but expand the range of future predictions to cover up to 120 min. The details of experimental parameters are provided in Appendix B.

The experiments were conducted on a desktop computer equipped with an NVIDIA GeForce RTX 3060 graphics processing unit, 16 GB of memory, and an Intel CPU i9-10900 K (3.70 GHz). All the baseline models were implemented or imported using Python from existing packages. The proposed model was implemented using PyTorch 1.9.0.

6. Results and analysis

6.1. Computational results

In this section, comparative experiments were conducted to verify the performance of the proposed models. ARIMA, KNN, MLP, LSTM, GRU, ASeq2Seq, DARNN, DCRNN, GWN, MTGNN, and ASTGNN were used as baseline models. The results of the one-step forecasting in terms of RMSE, MAE, and SMAPE are summarised in the Table 4, with the best performance marked in bold. Several interesting findings are as follows:

- Machine learning-based models such as KNN performed worse than other types of models, indicating the difficulty of traditional linear models in capturing the dynamics of short-term passenger flow.
- Models based on RNN architectures, including LSTM-FC and GRU-FC, which incorporate time-dependent modelling, outperformed KNN and MLP models in most tasks. GRU, as a variant of LSTM, showed comparable predictive power to LSTM.
- Incorporating the attention mechanism in the network, particularly the time attention mechanism, significantly improved the RNN's ability to capture time dependence. ASeq2Seq and DARNN, which utilise attention mechanisms, performed significantly better than single RNN architectures. e.g. LSTM and GRU.
- Embedding spatial modelling into short-term forecasting, e.g. GCRNN and GWN, effectively improved model performance by considering spatial dependence. Furthermore, graph learning-based models, which consider both temporal and spatial dependencies, outperformed the best DL models, i.e. ASeq2Seq and DARNN, that only focused on temporal dependency modelling.
- Models that incorporate both temporal and spatial attention mechanisms further enhanced the ability to capture temporal and spatial dependencies. ASTGNN, which includes both temporal and spatial attention mechanisms, outperformed models without attention mechanisms (GCRNN, GWN, and MTGNN) in most prediction tasks.
- Overall, our proposed approach, STA-PTCN-BiGRU, demonstrated the smallest RMSE, MAE, and SMAPE in all cases, indicating its effectiveness in short-term passenger flow prediction for metro systems.

Further analysis of our model's performance is presented in Table 5. The transfer type considers the average improvement of station S1 and S3, the non-transfer type considers the average improvement of station S2 and S4, and the average column indicates the average improvement of all four sample stations.

The average improvement can be observed in the 'Average' column of Table 5. Compared to the best performance of all ML-based models (ARIMA, KNN and MLP), our model shows average improvements of 27.32% in RMSE, 29.01% in MAE, and 38.23% in SMAPE. When compared to the best performance of all DL-based models (LSTM-FC, GRU-FC, ASeq2Seq and DARNN), our model achieves improvements of 36.53% in RMSE, 31.73% in MAE, and 29.65% in SMAPE. In comparison to the best performance achieved by all graph-based models (DCENN, GWN, MTGNN and ASTGNN), our model demonstrates improvements of 9.98% in RMSE, 8.08% in MAE, and 13.29% in SMAPE.

Our experiments confirm the effectiveness of the proposed model for both transfer-type and non-transfer-type stations. Specifically, in terms of the mean absolute error (MAE) metric, there is an improvement of 11.07% at transfer stations and 6.42% at non-transfer stations when compared to the best performance achieved by the baseline model, ASTGNN.

Table 4. Prediction error of the proposed method and the other baselines.

	S1(outflow)			S2(outflow)			S3(outflow)			S4(outflow)		
Methods	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE
ARIMA	54.04	39.19	23.52	69.35	43.15	24.56	50.27	35.33	51.18	41.88	25.83	54.42
KNN	71.28	44.87	22.31	91.47	49.40	23.29	66.30	40.45	48.54	55.24	29.57	51.62
MLP	69.63	45.90	25.07	89.35	50.54	26.18	64.77	41.38	54.55	53.96	30.25	58.01
LSTM-FC	65.95	41.95	22.81	84.62	46.26	23.81	61.34	38.04	49.62	51.10	27.89	52.77
GRU-FC	66.02	42.62	23.26	84.71	46.93	24.29	61.40	38.43	50.61	51.16	28.09	53.82
ASeq2Seq	61.89	40.75	19.59	79.41	44.86	20.45	57.56	36.74	42.62	47.96	26.85	45.32
DARNN	63.01	43.50	20.35	80.86	47.89	21.25	58.61	39.22	44.27	48.83	28.67	47.08
DCRNN	44.78	31.59	16.29	57.46	34.79	17.01	41.65	28.48	35.45	34.70	20.82	37.70
GWN	43.64	30.79	17.92	55.99	33.90	18.71	40.59	27.76	38.98	33.82	20.29	41.46
MTGNN	45.94	33.15	18.81	58.95	36.50	19.64	42.73	29.89	40.92	35.60	21.85	43.52
ASTGNN	43.65	30.26	15.89	56.01	33.32	16.59	40.60	27.28	34.58	33.83	19.94	36.77
Proposed	**39.42**	**27.72**	**13.46**	**49.35**	**31.27**	**14.77**	**37.72**	**24.09**	**30.24**	**30.17**	**18.78**	**31.57**

Table 5. The improvement of the proposed methods over baselines.

	Transfer type			Non-transfer type			Average		
Method	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE	RMSE	MAE	SMAPE
ARIMA	26.05%	30.47%	41.50%	28.51%	27.43%	41.33%	27.32%	29.01%	41.41%
KNN	78.36%	64.67%	62.13%	84.50%	57.78%	38.14%	44.90%	38.00%	38.23%
MLP	74.23%	68.47%	82.19%	80.22%	61.42%	44.96%	43.59%	39.40%	45.03%
LSTM-FC	65.00%	54.38%	65.74%	70.68%	48.17%	39.49%	40.44%	33.92%	39.57%
GRU-FC	65.18%	56.44%	69.04%	70.86%	49.89%	40.67%	40.50%	34.73%	40.76%
ASeq2Seq	54.85%	49.55%	42.35%	60.18%	43.29%	29.55%	36.53%	31.73%	29.65%
DARNN	57.66%	59.65%	47.88%	63.09%	52.97%	32.18%	37.66%	36.05%	32.28%
DCRNN	12.03%	15.96%	18.41%	15.89%	11.11%	15.30%	12.27%	11.95%	15.42%
GWN	9.18%	13.00%	30.20%	12.94%	8.27%	22.98%	9.98%	9.65%	23.08%
MTGNN	14.95%	21.69%	36.69%	18.91%	16.59%	26.63%	14.50%	16.10%	26.73%
ASTGNN	9.22%	11.07%	15.49%	12.98%	6.42%	13.17%	10.01%	8.08%	13.29%
Proposed	–	–	–	–	–	–	–	–	–

6.2. Multi-step prediction

To demonstrate the stability of our model, a multi-step forecasting task was conducted using datasets from four sample stations. The task involved predicting the outbound passenger flow for the next 3 (30 min), 6 (60 min), and 12 (120 min) target stations. ASeq2Seq, DCRNN, and ASTGNN were selected as benchmark models due to their superior multi-step prediction abilities.

The comparison of our method and baseline across different time lengths are shown in Figure 11. ‘ours’ in the legend represents the proposed model. As expected, the accuracy of predictions decreases as the length of the prediction time increases. However, our proposed model consistently outperforms the other three baseline models in all multi-step forecasting tasks. Notably, on the Station S4 (outflow) dataset, our model demonstrates a significant improvement over the baseline model ASTGNN. This improvement is particularly evident in the long-term prediction task (120 min), where our model exhibits stable prediction performance, which can be observed in Figure 11 (Station S4). Therefore, these results indicate that our proposed model is well-suited for relatively long prediction tasks and achieves satisfactory performance in the domain under study.

6.3. Ablation studies

In this section, the aim is to analyse the contributions of each module in our architecture, STA-PTCN-BiGRU. To facilitate this exploration, STA-PTCN-BiGRU is compared with its variants outlined below:

- w/o Cor-STFS (M1): This variant uses the raw data directly as input, without employing the Cor-STFS algorithm.
- w/o spatial and temporal attention (M2): In this variant, all attention modules are removed, retaining only the parallel TCN and BiGRU.
- w/o spatial attention (M3): This variant eliminates the spatial attention model, preserving only the attention module for temporal features.
- w/o temporal attention (M4): Conversely, this variant discards the temporal attention model, while maintaining the attention module for spatial features.

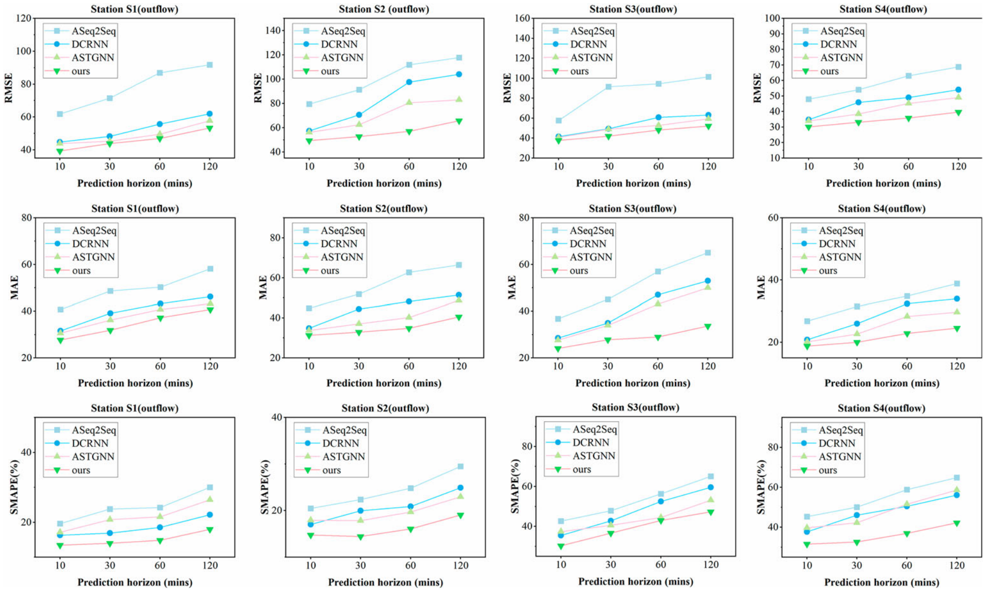

Figure 11. Performance comparison of our method and baselines as the prediction horizon increase.

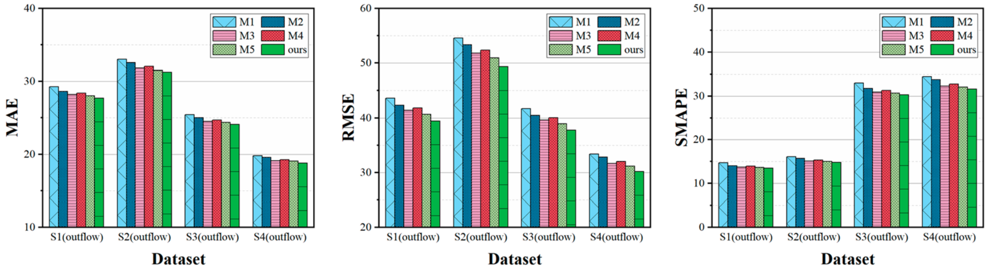

Figure 12. Ablation results on the variants of proposed method.

- w/o train timetable features (M5): This variant omits the train timetable features of the external event module within the model.

As shown in Figure 12, it can be observed that our model surpasses M1, M2, M3, M4, M5, and M6 in all test cases, indicating the positive impact of each module on the overall prediction accuracy.

Upon sophisticated data analysis, it is discovered that the utilisation of Cor-STFS results in an average RMSE reduction of 9.55%. This demonstrates that Cor-STFS eliminates interference from the original data and enhances the model's performance. Furthermore, employing the spatial-temporal attention module leads to an average RMSE reduction of 6.85%. Specifically, the inclusion of time attention reduces the RMSE by an average of 4.72%, while the incorporation of the spatial attention module reduces the RMSE by an average of 5.74%. These findings suggest that capturing temporal features is more crucial than capturing spatial features in the task of predicting passenger flow time series. Simultaneously considering

Table 6. The performance analysis of different spatial association thresholds (π) on the proposed method.

π	RMSE	MAE	SMAPE	Selected stations
0.95	41.75	29.68	14.83	13
0.90	40.23	28.30	13.85	40
0.85	**38.34**	**27.39**	**12.48**	**59**
0.80	39.42	27.72	13.46	80
0.75	40.41	29.07	15.22	94

temporal and spatial features enables the extraction of comprehensive spatial-temporal characteristics, thereby significantly improving prediction accuracy. Moreover, incorporating the train timetable feature within the external event module results in an average RMSE reduction of 3.09%. This underscores the effectiveness of considering the metro's inherent nature and extracting features from the timetable when predicting metro passenger flow.

Overall, our analysis highlights the significant contributions of each module to the model's accuracy. The Cor-STFS module improves performance by eliminating data interference, while the spatial-temporal attention, time attention, spatial attention, and train event feature modules further enhance prediction accuracy by capturing relevant features and characteristics of the metro system.

6.4. Influence of cor-sTFS

In this section, the effect of using the Corr-STFS algorithm on the prediction performance of the model was tested. Cor-STFS serves the purpose of selecting the optimal spatial association threshold and time-lag. The impact of these two values on the model's performance will be demonstrated step by step.

Spatial correlation plays a vital role in determining the input factors related to the spatial dimension. To illustrate this, let's consider the predicted target station S1. Tests were conducted to evaluate the effect of different choices of the spatial association threshold on the model, with a fixed time-lag of 8. The Poor forecast performance arises from overly pessimistic or strict values. This occurs when the spatial association threshold is excessively large, resulting in a small number of considered stations and insufficient spatial information related to the target station. Conversely, a small spatial association threshold incorporates more site-related information into the network, but it may introduce noise from other stations that are not highly correlated with the target station. This irrelevant data can adversely affect the model's inference and prediction, thereby undermining its performance. Based on the results presented in Table 6, it is found that a spatial association threshold of 0.85 yields the best prediction performance. In this case, 59 stations exhibiting high correlation are utilised as the spatial input for the model. This value was adopted as the default for the final model testing.

Time-lag is a crucial parameter for determining input factors related to the time dimension, as traffic dynamics exhibit strong correlation within nearby time periods. To examine the impact of different time-lag values while maintaining a spatial association threshold of 0.85, the RMSE, MAE, and SMAPE were assessed. As depicted in Figure 13, the results indicate that a time-lag of 8 yields the smallest errors in RMSE, MAE, and SMAPE. Interestingly, in the initial stages, the prediction error decreases rapidly as the time-lag increases. However, when the time-lag exceeds a certain critical point, the prediction error begins to increase

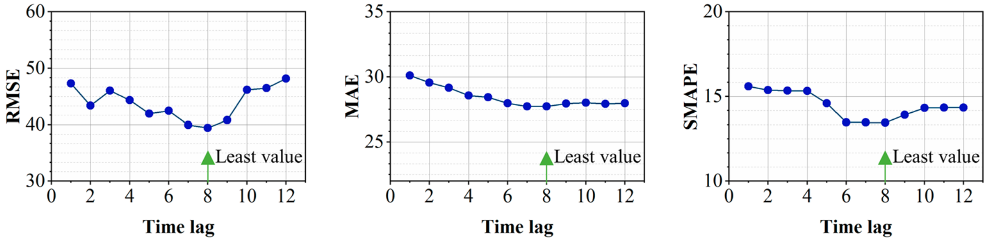

Figure 13. Influence of time lag on RMSE, MAE and SMAPE of the model when spatial association threshold equals 0.85.

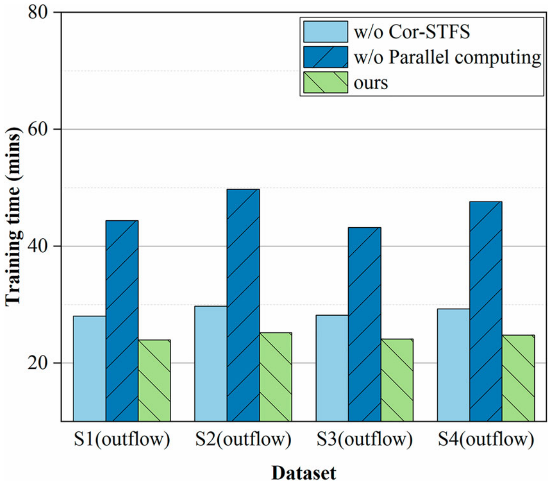

Figure 14. Effects of Cor-STFS and parallel computing on training time.

gradually. This can be attributed to the incorporation of excessively long temporal input, which complicates the forecasting model's ability to capture short-term dependencies.

6.5. Computational efficiency

In this section, tests were conducted to assess the impact of utilising Corr-STFS and parallel computing on the training time, as shown in Figure 14. Note that the 'w/o Cor-STFS' column implies using the original data directly as input without employing the Corr-STFS method. Similarly, the 'w/o parallel computing' column indicates reconstructing the proposed model by stacking temporal module and spatial module. These tests were performed using our approach and two variants on four case stations.

Our findings indicate that parallel computing offers the most substantial reduction in training time, with an average decrease of 46.98%. Corr-STFS also contributes to reducing training time, with an average decrease of 14.95%. This reduction can be attributed to Corr-STFS compressing the spatial-temporal features of the model, thereby leading to a decrease in the model's parameters.

randomness in passenger flow during weekends, which affects the model's stability. Similar trends are observed in the dataset for Station S2.

7. Conclusions

In this paper, we proposed an efficient parallel computing-based framework as an integral component of an intelligent transportation system for predicting metro passenger flow. Our framework incorporates relational information within the metro network, utilising the Cor-STFS algorithm for optimal input data selection and the STA-PTCN-BiGRU structure for capturing dynamic spatial-temporal characteristics. Through causal dilated convolution, residual connections, and bidirectional sequence information handling, our model effectively captures long-term dependencies and overcomes gradient problems associated with RNNs. External features and fully connected fusion layers are incorporated, providing comprehensive prediction results. Experiments using a real dataset from the Shanghai metro network were conducted to evaluate the accuracy and generality of our proposed method. Various baseline models were compared, and the results showcased the superior performance of our model. In terms of RMSE, we achieved improvements of 9.18% and 6.42% for transfer-type and non-transfer-type stations, respectively. Additionally, our model demonstrated outstanding stability and robustness in multi-step prediction tasks, outperforming all baseline models. The model ablation experiment highlighted the contributions of train event features and the Cor-STFS method, resulting in significant performance improvements. Moreover, the Cor-STFS method and parallel computing approach effectively reduced training time.

In real-world scenarios, the framework proposed in this paper plays a crucial role in planning train timetables and facilitating real-time operational rescheduling. It effectively addresses the spatial-temporal imbalance in passenger demand by adjusting train stop plans and introducing additional trains in daily operations. Moreover, by taking into account station ridership, the framework aids in improving in-station service facilities and mitigating potential congestion through the implementation of emergency protective measures.

However, some areas warrant further exploration in this paper. Our study does not account for abnormal passenger flow during major incidents, as explored by Pasini et al. (2022). This limitation can be addressed in future work by investigating the adaptability of our model for changing passenger conditions, such as dynamics during planned and unplanned disruptions. Furthermore, we plan to investigate the adaptability of various correlation methods, such as cross-correlation coefficients, and their potential application in metro passenger flow prediction for our future research.

Acknowledgements

This work was supported by grants from the National Engineering Laboratory of Integrated Transportation Big Data Application Technology, and National and Local Joint Engineering Laboratory of Comprehensive Intelligent Transportation, Chengdu, Sichuan, China. The work of Shuguang Zhan was supported in part by the National Natural Science Foundation of China under Grant 72171198 and Natural Science Foundation of Sichuan Province under Grant 2022NSFSC1890. The work of Qiyuan Peng was supported by the High-speed Railway Basic Research Joint Fund of China under Grant U1834209. The work of S.C. Wong was supported by Francis S Y Bong Endowed Professorship in Engineering.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by National Natural Science Foundation of China: [Grant Number 72171198]; Natural Science Foundation of Sichuan Province: [Grant Number 2022NSFSC1890]; Francis S Y Bong Endowed Professorship in Engineering: [Grant Number]; Guangdong - Hong Kong - Macau Joint Laboratory Program of the 2020 Guangdong New Innovative Strategic Research Fund, Guangdong Science and Technology Department: [Grant Number 2020B1212030009]; National Engineering Laboratory of Integrated Transportation Big Data Application Technology: [Grant Number]; the High-speed Railway Basic Research Joint Fund of China: [Grant Number U1834209]; National and Local Joint Engineering Laboratory of Comprehensive Intelligent Transportation: [Grant Number].

ORCID

Cong Xiu http://orcid.org/0000-0002-6393-7883
Shuguang Zhan http://orcid.org/0000-0001-8252-2782

References

Aboudolas, K., M. Papageorgiou, and E. Kosmatopoulos. 2009. "Store-and-forward Based Methods for the Signal Control Problem in Large-Scale Congested Urban Road Networks." *Transportation Research Part C: Emerging Technologies* 17: 163–174. https://doi.org/10.1016/j.trc.2008.10.002.

Ahn, J., E. Ko, and E. Y. Kim. 2016. "Highway Traffic Flow Prediction Using Support Vector Regression and Bayesian Classifier." 2016 Int. Conf. Big Data Smart Comput. Big Comp 2016, 239–244. https://doi.org/10.1109/BIGCOMP.2016.7425919

Arroyo, J., and C. Maté. 2009. "Forecasting Histogram Time Series with k-Nearest Neighbours Methods." *International Journal of Forecasting* 25: 192–207. https://doi.org/10.1016/j.ijforecast.2008.07.003.

Bahdanau, D., K. H. Cho, and Y. Bengio. 2014. "Neural Machine Translation by Jointly Learning to Align and Translate." 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc. https://doi.org/10.48550/arxiv.1409.0473

Bai, S., J. Z. Kolter, and V. Koltun. 2018. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling.

Belletti, F., D. Haziza, G. Gomes, and A. M. Bayen. 2018. "Expert Level Control of Ramp Metering Based on Multi-Task Deep Reinforcement Learning." *IEEE Transactions on Intelligent Transportation Systems* 19: 1198–1207. https://doi.org/10.1109/TITS.2017.2725912.

Castro-Neto, M., Y.-S. Jeong, M.-K. Jeong, and L. D. Han. 2009. "Online-SVR for Short-Term Traffic Flow Prediction Under Typical and Atypical Traffic Conditions." *Expert Systems with Applications* 36: 6164–6173. https://doi.org/10.1016/j.eswa.2008.07.069.

Chen, B. Y., Y. Ma, J. Wang, T. Jia, X. Liu, and W. H. K. Lam. 2023. "Graph Convolutional Networks with Learnable Spatial Weightings for Traffic Forecasting Applications." *Transportmetrica A: Transport Science*, 1–30. https://doi.org/10.1080/23249935.2023.2239377.

Cho, K., B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. "Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation." *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, https://doi.org/10.3115/v1/d14-1179.

Duan, Y., Y. Lv, and F. Y. Wang. 2016. "Travel Time Prediction with LSTM Neural Network." IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC. Institute of Electrical and Electronics Engineers Inc., 1053–1058. https://doi.org/10.1109/ITSC.2016.7795686

Fu, R., Z. Zhang, and L. Li. 2016. "Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction." Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016. Institute of Electrical and Electronics Engineers Inc., 324–328. https://doi.org/10.1109/YAC.2016.7804912

Geng, X., Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu. 2019. "Spatiotemporal Multi-Graph Convolution Network for Ride-Hailing Demand Forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence* 33: 3656–3663. https://doi.org/10.1609/aaai.v33i01.33013656.

Guo, S., Y. Lin, N. Feng, C. Song, and H. Wan. 2019. "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting." *Proceedings of the AAAI Conference on Artificial Intelligence*, 922–929. https://doi.org/10.1609/aaai.v33i01.3301922.

Guo, S., Y. Lin, H. Wan, X. Li, and G. Cong. 2021. "Learning Dynamics and Heterogeneity of Spatial-Temporal Graph Data for Traffic Forecasting." *IEEE Transactions on Knowledge and Data Engineering* 34: 5415–5428. https://doi.org/10.1109/TKDE.2021.3056502.

Guo, J., Y. Liu, Q. Yang, Y. Wang, and S. Fang. 2021. "GPS-based Citywide Traffic Congestion Forecasting Using CNN-RNN and C3D Hybrid Model." *Transportmetrica A: Transport Science* 17: 190–211.

Guo, J., W. Wang, Y. Tang, Y. Zhang, and H. Zhuge. 2022. "A CNN-Bi_LSTM Parallel Network Approach for Train Travel Time Prediction." *Knowledge-Based Systems* 256: 109796. https://doi.org/10.1016/j.knosys.2022.109796.

Hao, S., D. H. Lee, and D. Zhao. 2019. "Sequence to Sequence Learning with Attention Mechanism for Short-Term Passenger Flow Prediction in Large-Scale Metro System." *Transportation Research Part C: Emerging Technologies* 107: 287–300. https://doi.org/10.1016/j.trc.2019.08.005.

He, Y., Y. Zhao, and K.-L. Tsui. 2023. "Short-term Forecasting of Origin-Destination Matrix in Transit System via a Deep Learning Approach." *Transportmetrica A: Transport Science* 19: 2033348.

Jiang, Y., S. Gao, W. Guan, and X. Yin. 2022. "Bass+ BL+ Seasonality Forecasting Method for Demand Trends in air Rail Integrated Service." *Transportmetrica A: Transport Science* 18: 281–298.

Ke, J., H. Zheng, H. Yang, and X. Chen. 2017. "Short-Term Forecasting of Passenger Demand Under On-Demand Ride Services: A Spatio-Temporal Deep Learning Approach." *Transportation Research Part C: Emerging Technologies* 85: 591–608. https://doi.org/10.1016/j.trc.2017.10.016.

Kim, T., S. Sharda, X. Zhou, and R. M. Pendyala. 2020. "A Stepwise Interpretable Machine Learning Framework Using Linear Regression (LR) and Long Short-Term Memory (LSTM): City-Wide Demand-Side Prediction of Yellow Taxi and for-Hire Vehicle (FHV) Service." *Transportation Research Part C: Emerging Technologies* 120: 102786. https://doi.org/10.1016/j.trc.2020.102786.

Kumar, S. V. 2017. "Traffic Flow Prediction Using Kalman Filtering Technique." *Procedia Engineering* 187: 582–587. https://doi.org/10.1016/j.proeng.2017.04.417.

Lee, S., and D. B. Fambro. 1999. "Application of Subset Autoregressive Integrated Moving Average Model for Short-Term Freeway Traffic Volume Forecasting." *Transportation Research Record: Journal of the Transportation Research Board* 1678: 179–188. https://doi.org/10.3141/1678-22.

Lee, K., and W. Rhee. 2022. "DDP-GCN: Multi-Graph Convolutional Network for Spatiotemporal Traffic Forecasting." *Transportation Research Part C: Emerging Technologies* 134: 103466. https://doi.org/10.1016/j.trc.2021.103466.

Li, Y., R. Yu, C. Shahabi, and Y. Liu. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv Prepr. arXiv1707.01926.

Li, C., L. Zheng, and N. Jia. 2024. "Network-wide Ride-Sourcing Passenger Demand Origin-Destination Matrix Prediction with a Generative Adversarial Network." *Transportmetrica A: Transport Science* 20 (1): 2109774. https://doi.org/10.1080/23249935.2022.2109774..

Lin, Z., J. Feng, Z. Lu, and Y. Li. 2019. Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis. ojs.aaai.org 19.

Liu, L., J. Chen, H. Wu, J. Zhen, G. Li, and L. Lin. 2022. "Physical-Virtual Collaboration Modeling for Intra- and Inter-Station Metro Ridership Prediction." *IEEE Transactions on Intelligent Transportation Systems*, https://doi.org/10.1109/TITS.2020.3036057.

Liu, Y., Z. Liu, and R. Jia. 2019. "DeepPF: A Deep Learning Based Architecture for Metro Passenger Flow Prediction." *Transportation Research Part C: Emerging Technologies* 101: 18–34. https://doi.org/10.1016/j.trc.2019.01.027.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F.-Y. Wang. 2014. "Traffic Flow Prediction with Big Data: A Deep Learning Approach." *IEEE Transactions on Intelligent Transportation Systems* 16: 865–873.

Lv, M., Z. Hong, L. Chen, T. Chen, T. Zhu, and S. Ji. 2021. "Temporal Multi-Graph Convolutional Network for Traffic Flow Prediction." *IEEE Transactions on Intelligent Transportation Systems*, https://doi.org/10.1109/TITS.2020.2983763.

Ma, X., Z. Dai, Z. He, J. Ma, Yong Wang, and Yunpeng Wang. 2017. “Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction.” *Sensors (Switzerland)* 17. https://doi.org/10.3390/s17040818.

Ma, X., Z. Tao, Yinhai Wang, H. Yu, and Yunpeng Wang. 2015. “Long Short-Term Memory Neural Network for Traffic Speed Prediction Using Remote Microwave Sensor Data.” *Transportation Research Part C: Emerging Technologies* 54: 187–197. https://doi.org/10.1016/j.trc.2015.03.014.

Ma, C., Y. Zhao, G. Dai, X. Xu, and S.-C. Wong. 2022. “A Novel STFSA-CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction.” *IEEE Transactions on Intelligent Transportation Systems*.

Nagy, A. M., and V. Simon. 2018. “Survey on Traffic Prediction in Smart Cities.” *Pervasive and Mobile Computing* 50: 148–163. https://doi.org/10.1016/j.pmcj.2018.07.004.

Pasini, K., M. Khouadjia, A. Samé, M. Trépanier, and L. Oukhellou. 2022. “Contextual Anomaly Detection on Time Series: A Case Study of Metro Ridership Analysis.” *Neural Computing and Applications*, 1–25.

Sattarzadeh, A. R., R. J. Kutadinata, P. N. Pathirana, and V. T. Huynh. 2023. “A Novel Hybrid Deep Learning Model with ARIMA Conv-LSTM Networks and Shuffle Attention Layer for Short-Term Traffic Flow Prediction.” *Transportmetrica A: Transport Science*, 1–23. https://doi.org/10.1080/23249935.2023.2236724.

Shahriari, S., M. Ghasri, S. A. Sisson, and T. Rashidi. 2020. “Ensemble of ARIMA: Combining Parametric and Bootstrapping Technique for Traffic Flow Prediction.” *Transportmetrica A: Transport Science* 16: 1552–1573.

Sheu, J.-B., L. W. Lan, and Y.-S. Huang. 2009. “Short-term Prediction of Traffic Dynamics with Real-Time Recurrent Learning Algorithms.” *Transportmetrica A: Transport Science* 5: 59–83.

Shi, Z., N. Zhang, P. M. Schonfeld, and J. Zhang. 2020. “Short-term Metro Passenger Flow Forecasting Using Ensemble-Chaos Support Vector Regression.” *Transportmetrica A: Transport Science* 16: 194–212.

Tang, J., S. Alelyani, and H. L. And. 2014. “Feature Selection for Classification: A Review.” *Data Classif. Appl* 37.

Tang, J., J. Liang, F. Liu, J. Hao, and Y. Wang. 2021. “Multi-community Passenger Demand Prediction at Region Level Based on Spatio-Temporal Graph Convolutional Network.” *Transportation Research Part C: Emerging Technologies* 124: 102951. https://doi.org/10.1016/j.trc.2020.102951.

Tedjopurnomo, D. A., Z. Bao, B. Zheng, F. M. Choudhury, and A. K. Qin. 2020. “A Survey on Modern Deep Neural Network for Traffic Prediction: Trends, Methods and Challenges.” *IEEE Transactions on Knowledge and Data Engineering* 34: 1544–1561.

Vaswani, A., G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł Kaiser, and I. Polosukhin. 2017. Attention is all You Need. proceedings.neurips.cc.

Wei, P., Y. Cao, and D. Sun. 2013. “Total Unimodularity and Decomposition Method for Large-Scale air Traffic Cell Transmission Model.” *Transportation Research Part B: Methodological* 53: 1–16. https://doi.org/10.1016/j.trb.2013.03.004.

Williams, B. M. 2001. “Multivariate Vehicular Traffic Flow Prediction: Evaluation of ARIMAX Modeling.” *Transportation Research Record: Journal of the Transportation Research Board*, 194–200. https://doi.org/10.3141/1776-25.

Williams, B. M., and L. A. Hoel. 2003. “Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results.” *Journal of Transportation Engineering* 129: 664–672. https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(664).

Wu, C. H., J. M. Ho, and D. T. Lee. 2004. “Travel-time Prediction with Support Vector Regression.” In *IEEE Transactions on Intelligent Transportation Systems*, 276–281. https://doi.org/10.1109/TITS.2004.837813

Wu, Z., S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang. 2020. “Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks.” Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 753–763.

Wu, Z., S. Pan, G. Long, J. Jiang, and C. Zhang. 2019. Graph Wavenet for Deep Spatial-Temporal Graph Modeling. arXiv Prepr. arXiv1906.00121.

Xiu, C., Y. Sun, and Q. Peng. 2022a. “Modelling Traffic as Multi-Graph Signals: Using Domain Knowledge to Enhance the Network-Level Passenger Flow Prediction in Metro Systems.” *Journal of Rail Transport Planning & Management* 24: 100342.

Xiu, C., Y. Sun, Q. Peng, C. Chen, and X. Yu. 2022b. "Learn Traffic as a Signal: Using Ensemble Empirical Mode Decomposition to Enhance Short-Term Passenger Flow Prediction in Metro Systems." *Journal of Rail Transport Planning & Management* 22: 100311.

Xue, G., S. Liu, L. Ren, Y. Ma, and D. Gong. 2022. "Forecasting the Subway Passenger Flow Under Event Occurrences with Multivariate Disturbances." *Expert Systems with Applications* 188: 116057. https://doi.org/10.1016/j.eswa.2021.116057.

Yang, S. 2013. "On Feature Selection for Traffic Congestion Prediction." *Transportation Research Part C: Emerging Technologies* 26: 160–169. https://doi.org/10.1016/j.trc.2012.08.005.

Yao, H., F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, D. Chuxing, and Z. Li. 2018. Deep Multi-View Spatial-Temporal Network for Taxi Demand Prediction. ojs.aaai.org.

Ye, J., J. Zhao, K. Ye, and C. Xu. 2022. "How to Build a Graph-Based Deep Learning Architecture in Traffic Domain: A Survey." *IEEE Transactions on Intelligent Transportation Systems* 23: 3904–3924. https://doi.org/10.1109/TITS.2020.3043250.

Yu, H., N. Ji, Y. Ren, and C. Yang. 2019. "A Special Event-Based K-Nearest Neighbor Model for Short-Term Traffic State Prediction." *IEEE Access* 7: 81717–81729. https://doi.org/10.1109/ACCESS.2019.2923663.

Yu, H., Z. Wu, S. Wang, Y. Wang, and X. Ma. 2017. "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks." *Sensors (Switzerland)* 17. https://doi.org/10.3390/s17071501.

Yuan, J., Y. Zheng, X. Xie, and G. Sun. 2011. "Driving with Knowledge from the Physical World." Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 316–324. https://doi.org/10.1145/2020408.2020462.

Zeng, J., and J. Tang. 2023. "Combining Knowledge Graph Into Metro Passenger Flow Prediction: A Split-Attention Relational Graph Convolutional Network." *Expert Systems with Applications* 213: 118790. https://doi.org/10.1016/j.eswa.2022.118790.

Zhang, J., F. Chen, Z. Cui, Y. Guo, and Y. Zhu. 2020. "Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit." *IEEE Transactions on Intelligent Transportation Systems*, 7004–7014. https://doi.org/10.1109/TITS.2020.3000761.

Zhang, Z., M. Li, X. Lin, Y. Wang, and F. He. 2019a. "Multistep Speed Prediction on Traffic Networks: A Deep Learning Approach Considering Spatio-Temporal Dependencies." *Transportation Research Part C: Emerging Technologies* 105: 297–322. https://doi.org/10.1016/j.trc.2019.05.039.

Zhang, Y., and Y. Liu. 2009. "Traffic Forecasting Using Least Squares Support Vector Machines." *Transportmetrica A: Transport Science* 5: 193–213.

Zhang, J., F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen. 2011. "Data-driven Intelligent Transportation Systems: A Survey." *IEEE Transactions on Intelligent Transportation Systems* 12: 1624–1639. https://doi.org/10.1109/TITS.2011.2158001.

Zhang, W., Y. Yu, Y. Qi, F. Shu, and Y. Wang. 2019b. "Short-term Traffic Flow Prediction Based on Spatio-Temporal Analysis and CNN Deep Learning." *Transportmetrica A: Transport Science* 15: 1688–1711. https://doi.org/10.1080/23249935.2019.1637966.

Zhang, J., Y. Zheng, and D. Qi. 2017. "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction." 31st AAAI Conference on Artificial Intelligence, AAAI 2017. AAAI Press, pp. 1655–1661.

Zhao, L., Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li. 2020. "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction." *IEEE Transactions on Intelligent Transportation Systems* 21: 3848–3858. https://doi.org/10.1109/TITS.2019.2935152.

Zheng, C., X. Fan, C. Wang, and J. Qi. 2020. "Gman: A Graph Multi-Attention Network for Traffic Prediction." *Proceedings of the AAAI Conference on Artificial Intelligence* 34 (01): 1234–1241. https://doi.org/10.1609/aaai.v34i01.5477.

Appendix A. Time-of-day factor and day-of-week factor

As shown of **Table A.7**, the time-of-day factors is divided into the following 7 categories according to the general Chinese metro travel mode. As the time-of-day variable is also a discrete variable, the one-hot method can be used to encode it. For example, the evening peak time period can be expressed as $[0, 0, 0, 0, 0, 1, 0]$.

The day-of-week factor can be simply divided into two categories, including the workday mode and the weekend mode. Similar to time-of-day variable, one-hot representation can also be used to describe it. For example, workdays can be denoted as $[1, 0]$.

Table A7. Time period classification.

Category index	Period	Description
1	6 am —7 am	Pre morning peak
2	7 am - 9 am	Morning peak
3	9 am - 10 am	End morning peak
4	10 am - 16 pm	Flat hump period
5	16 pm - 17 pm	Pre evening peak
6	17 pm - 20 pm	Evening peak
7	20 pm - 23 pm	End Evening peak

Appendix B. Experimental parameters

In this section, we present the parameter settings and details of the proposed approach. The parameter settings for the proposed method, including parameters for the neural network, optimiser, and spatiotemporal feature selection, are presented in **Table B.8**. Via conducting numerical experiments and fine-tuning, the best network configuration that yields the lowest error in both the training and validation sets can be obtained. The best configuration consists of a 3-layer TCN module with an optimal kernel size of 3, and the BiGRU module has 64 units in each direction. The details of the proposed architecture used in this paper are shown in **Table B.9**. Furthermore, the Cor-STFS feature selection method resulted in a spatial association threshold value of 0.85 and a time lag value of 8.

Table B8. Parameter settings for the proposed method.

Items	Settings
TCN layers	[1,5]
TCN kernel size	[2,7]
Initial learning rate	0.001
Batch_size	128
Dropout	0.2
Optimiser	Adam
Loss function	MAE
Spatial association threshold	0.75,0.80,0.85,0.90,0.95
Time lag	[1,12]

Table B9. Details of the proposed architecture.

Layer	Hyperparameter	value
Causal dilated conv layer 1	Dilation factor	1
	Convolution kernels	64
	Kernel size	3
Causal dilated conv layer 2	Dilation factor	2
	Convolution kernels	64
	Kernel size	3
Causal dilated conv layer 3	Dilation factor	4
	Convolution kernels	32
	Kernel size	3
BiGRU	Each direction unit size	64

Appendix C. Visual interpretation

In this section, the one-week prediction results of the proposed method on the test set were visualised. **Figure C.15** displays the predicted values alongside the corresponding observed values. It can be observed that the predicted values align with the observed trend for the target station, thus confirming the model's ability to capture spatial-temporal dynamics of passenger flow. Furthermore, the visualised results for both transfer station types (S1, S3) and non-transfer station types (S2, S4) demonstrate the reliability of the proposed model, with prediction errors being within an acceptable range. To further examine the performance of the proposed method on workdays and weekends, Station S1 (transfer-type) and Station S2 (non-transfer type) were chosen as examples and visualised the predictive performance for both Monday and Saturday. The detailed prediction results of Station S1 and S2 are shown in **Figure C.16** and **Figure C.17**. It is found that the proposed model effectively captures traffic bursts in passenger flow on both workdays and weekends at Station S1. However, the model exhibits better performance on workdays compared to weekends. This disparity may be attributed to the increased randomness in passenger flow during weekends, which affects the model's stability.

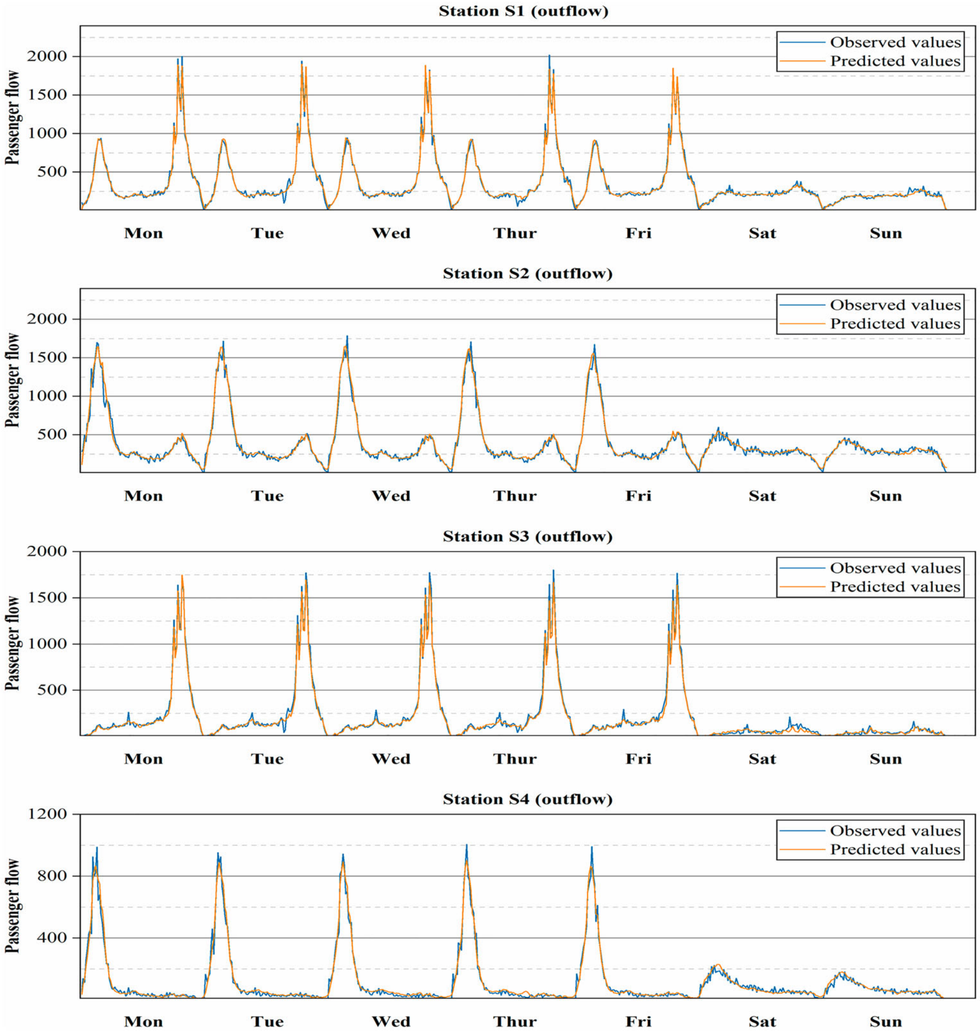

Figure C15. Prediction results compared with observations on all tested datasets.

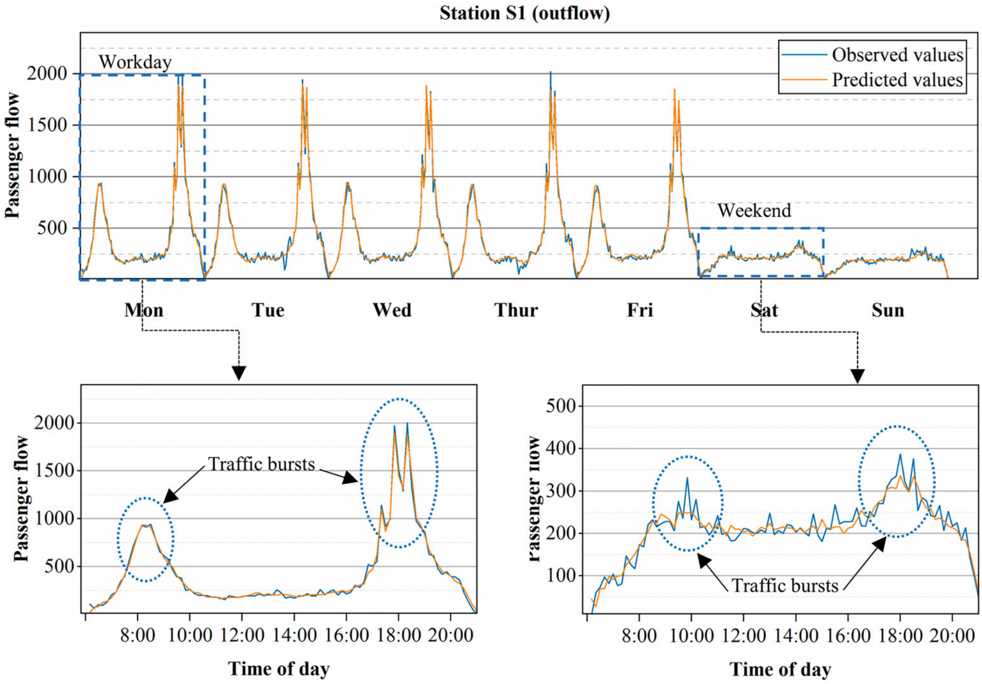

Figure C16. The detailed prediction result of passenger flow at Station S1 using the proposed method.

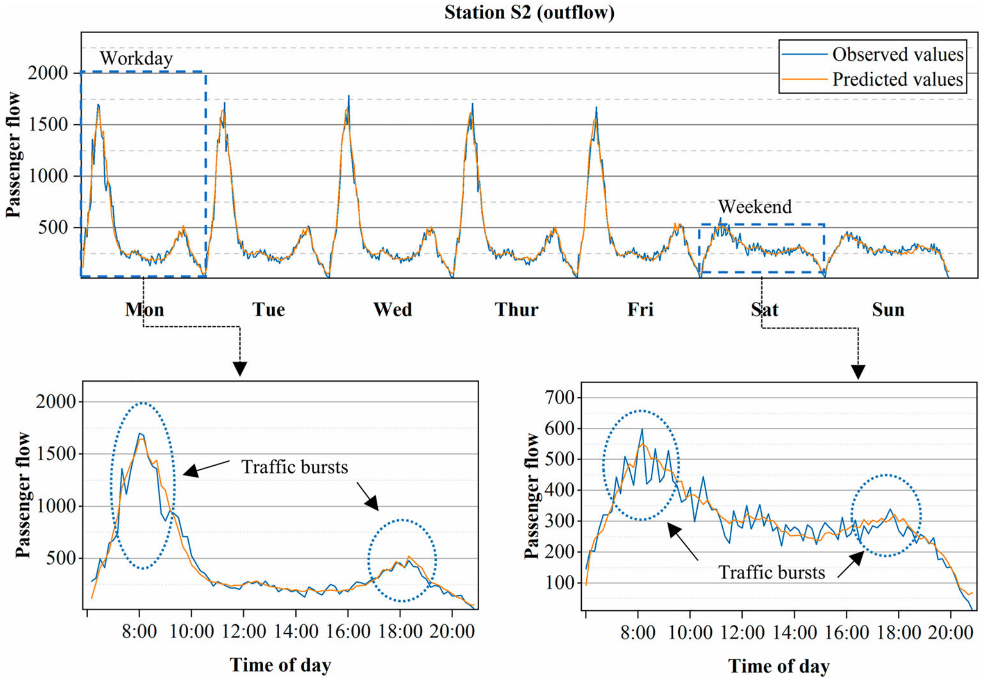

Figure C17. The detailed prediction result of passenger flow at Station S2 using the proposed method.