ITM Web of Conferences **84**, 03019 (2026) https://doi.org/10.1051/itmconf/20268403019
ATCIDS 2026

Short-term Passenger Flow Prediction and Operation Optimization for Urban Rail Transit Based on Multi-model Comparison

Sicai Zhai[1*]

[1]College of Big Data and Artificial Intelligence, Guangdong University of Finance, Qingyuan, 511510, China

Abstract. With the continuous growth of public transportation demand in large cities, metro operations are facing increasingly severe passenger flow pressure. Accurate short-term passenger flow forecasting is very important. It helps transportation systems run smoothly. Passenger experience improves because of it. Operational management also gets better. Based on Shenzhen's subway passenger flow and meteorological data, this paper compares multiple models. The goal is to forecast short-term subway passenger flow. Weather factors affect passenger flow. The paper analyzed this effect separately, it is found that rainfall and strong winds significantly reduce passenger numbers, while high temperatures lead to an increase. The study compares the predictive performance of three models—CNN-LSTM, XGBoost, and STGCN—revealing that STGCN outperforms in MAE, RMSE, MAPE, and R^2 metrics, effectively capturing spatiotemporal dependencies; XGBoost offers high training efficiency, making it suitable for real-time scenarios; and CNN-LSTM demonstrates strong trend fitting. This research helps with model selection. Different cities can use these criteria for passenger flow forecasting. The findings also support operations. They provide data for optimizing subway management.

1 Introduction

With the continuous acceleration of global urbanization, subway systems have become the backbone of public transportation in large and medium-sized cities, handling tens of millions of passengers daily. However, the rapidly growing passenger demand has brought unprecedented pressure to subway operations, especially during morning and evening peak hours. Issues such as overcrowded platforms and train cars, and train delays are becoming increasingly prominent, which not only severely affect passengers' travel experience and efficiency but also pose potential threats to public safety. Subway passenger flow shows strong correlation over time. This means past data influences future flow. Therefore, a prediction model must use historical data. Understanding the dynamic information of subway passenger flow changes is necessary. Tracking how passenger flow changes over time is the specific method. A data-driven approach can optimize prediction outcomes. This method

* Corresponding author's email: Zsc15816343828@outlook.com

improves both the timeliness and the accuracy of predictions [1], achieving intelligent and refined management of subway operations, and shifting from passive response to proactive intervention, has become a key challenge that urgently needs to be addressed in the field of urban traffic management.

Current subway passenger flow research often focuses on one city or one model. Models like ARIMA or LSTM are common examples. This focus has two main problems. First, a single model struggles to adapt. Different cities have different subway network layouts. Their passenger flow also varies in time and space. Second, multi-city comparisons are rare. The performance of various models across cities is not compared. This gap means a clear "model selection guide" for different cities cannot be provided. Weather factors can affect passenger flow. Existing studies often miss this. For example, rain or high temperatures matter. They can make commuters leave earlier. They can also cut down on non-essential travel. Prediction accuracy then drops in extreme weather. But these models are not useless. They can still give good help sometimes. When predicting subway station entry volumes, the seasonal ARIMA model can effectively describe trends and inherent periodic characteristics of passenger flow data. Seasonal adjustment can remove seasonal variation elements in the entry volume time series, thereby revealing the underlying trend and cyclical components of the series, to realistically reflect the periodic characteristics of passenger flow and the patterns of the time series [2]. LSTM can be used for data preprocessing and can build multivariate time series models to predict subway passenger flow [3].

Linear prediction works with simple rules. Rule one: the data should be in a straight line. Rule two: the data should not change its pattern over time. The method looks at old data. It sees how the data changed before. Then, it uses that old pattern to guess the future. Machine learning models can predict things more accurately. Support Vector Machines (SVM) and random forests are examples of such models. Lam and others have shown that when predicting daily traffic flow in Hong Kong, SVM is a good predictive model [4]. Machine learning models enhance nonlinear fitting capabilities through feature mapping, continuously learning the characteristics present in the data, thereby achieving a mapping relationship between inputs and optimal crowd flow control [5]. However, most machine learning models cannot account for the more complex spatiotemporal correlations between stations, deep learning models capture time-based patterns well. Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) are two such models. Because of this strength, they are now the most common method used. LSTM models predict passenger flow well. They work for different kinds of subway stations. These models offer both theory and data for planning. They help forecast how crowded stations will get. They help show future passenger trends. This aids in organizing crowds. It helps manage passenger movement. The models also assist with emergency risks. Overall, they improve service quality. They make subway operations safer [6]. Since deep learning models can more accurately describe the complex relationships between input and output layers, researchers have tried to solve traffic flow problems. They use deep learning models for this. These models aim to forecast traffic flow [7], which also greatly aids in short-term predictions of subway passenger flow.

Therefore, this study tests several prediction models. The models include CNN-LSTM, XGBoost, and STGCN. The test will be careful and organized. It checks how well each model forecasts short-term metro passenger flow. This check happens across many different cities. The goal is to see the performance differences between the models, explore the impact mechanisms of meteorological factors on passenger flow, and propose model selection strategies suitable for different operational scenarios. The ultimate goal is to provide subway operations management departments with an efficient and accurate method for short-term passenger flow prediction, supporting dynamic scheduling and optimized resource allocation, and enhancing the intelligent operation level of urban rail transit systems.

2 Related work

2.1 Data source

The data for this study comes from an open source — Shenzhen Open Data Platform, covering the period from December 3, 2022, to December 3, 2024. This data includes passenger flow in the Shenzhen Metro, combined with meteorological data from WeatherArchive, to analyze the impact of weather changes on Shenzhen Metro passenger flow.

2.2 Data analysis

This study examines the daily passenger flow of the Shenzhen Metro, dividing it into normal commuting weekdays (Monday to Friday) and non-normal commuting times on weekends (Saturday and Sunday), and studies their relationship with various weather factors. The relationships between different weather factors and the daily metro passenger flow f (unit: ten thousand people) are analyzed separately for weekdays and weekends. Then, the number of days d is extracted to obtain the variance S^2 and the significance indicator p, as shown in Table 1. Rainy days are categorized into light rain (Class I) and heavy rain (Class II), wind force is divided into levels 0-4 and 5-8, and temperature is divided into >30℃ and 0-30℃.

Table 1. Statistical analysis of daily subway passenger flow and weather factors on weekdays and weekends.

Weather factors		Workday				Weekend			
		Number of days/d	Workday/f	S^2	p	Number of days/d	Weekend/f	S^2	p
Weather conditions	I Category	427	918.4	5832.6	0.000	161	631.5	3197.4	0.000
	II Category	92	860.2	4106.7	—	38	598.7	2189.5	—
Wind power	0-4 Level	404	923.5	5694.3	0.021	154	636.8	3251.6	0.021
	5-8 Level	115	887.3	5218.9	—	45	609.4	2674.2	—
Temperature	>30℃	329	901.6	5381.4	0.000	124	621.3	3066.7	0.000
	0-30℃	190	941.2	4912.5	—	75	654.8	2987.1	—

At the same time, the passenger flow on each line was analyzed, with the time period from March 1, 2023, to May 31, 2023 (Figure 1).

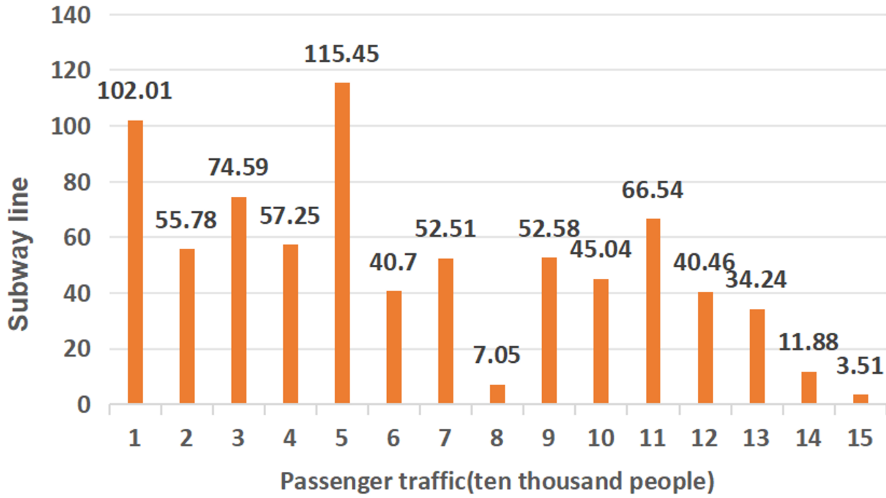

Fig. 1. Passenger traffic on various lines in Shenzhen over the past three months (Picture credit: Original)

2.3 The results show

In Table 1, there are significant differences in daily passenger flow between weather conditions Type I and Type II on both weekdays and weekends ($p<0.001$). Rainy days significantly reduce passenger flow for both Type I and Type II conditions, with a decrease of 6.3% on weekdays and about 5.2% on weekends. Strong winds also result in a decrease in passenger flow, with significant differences ($p=0.021$), showing a reduction of about 3.9% on weekdays and about 4.3% on weekends. High temperatures (>30℃) significantly increase passenger flow, with an increase of about 4.4% on weekdays and about 5.4% on weekends ($p<0.001$). The significant drop in passenger flow during rainy and windy days may be because people are less willing to go out, leading to lower ridership. On the other hand, during high-temperature weather, there is the normal demand for commuting on weekdays. The increase in subway passenger flow can be attributed partly to the fact that the subway provides a cool and comfortable environment during hot weather, allowing travelers to reach their destinations quickly and on time, thereby attracting those who might otherwise walk or cycle short distances. Additionally, compared to other medium- and long-distance travel options such as buses, taxis, or private cars, the subway is more comfortable and economical in hot weather, resulting in a significant increase in the number of subway passengers [8].

In Figure 1, the passenger flow on lines 5, 11, 1, and 3 is relatively high compared to all other lines in Shenzhen. This indirectly reflects that these lines pass through Shenzhen's commercial and entertainment centers. At the same time, it can be analyzed that policymakers can allocate more resources by providing more public transportation infrastructure in these areas to achieve a better travel experience [9].

2.4 Method introduction

Long Short-Term Memory is called LSTM. It is a major type of Recurrent Neural Network (RNN). Its design fixes a common issue in standard RNNs. This issue is the vanishing gradient problem. The fix lets the model remember information from much earlier in a sequence. Therefore, LSTM works very well for data where order matters. Long Short-Term Memory (LSTM) networks improve on basic RNNs. They add a special part called a memory cell. This cell is inside the hidden layer of the network. Its job is to let the network remember information for a long time. This gives RNNs the ability for long-term memory. LSTM uses three controllable gates to weigh historical and current information, optimizing and producing the best model. The LSTM control gates are illustrated in Figure 1. Based on RNNs, LSTM calculates the current cell state based on the previous temperature output and the current temperature input. Through the control of the forget gate, the new cell state can retain information from long ago, and through the control of the input gate, it can prevent unimportant features at the current time step from entering the memory cell.

CNN is a deep learning model specifically designed to process data with network structures (such as images and speech spectrograms). Its core advantage is efficiently extracting spatial features of the data through local perception and parameter sharing, significantly reducing computational cost. Figure 2 shows the structure of a CNN. The VGG-16 model has a key feature. This feature is the use of 'small convolutional kernels stacked in multiple layers.' Specifically, it uses many small 3x3 filters. It does not use large filters. This design choice is central to its core idea. This method has two main benefits. First, it allows every output pixel to receive information from a specific area of the input. Second, it lowers the total number of parameters the model needs. These two benefits also help the model learn more complex patterns. The repeated convolution-pooling structure allows the model to extract features layer by layer from 'low-level textures to high-level semantics, making it very suitable for tasks such as subway image classification and passenger flow target detection.

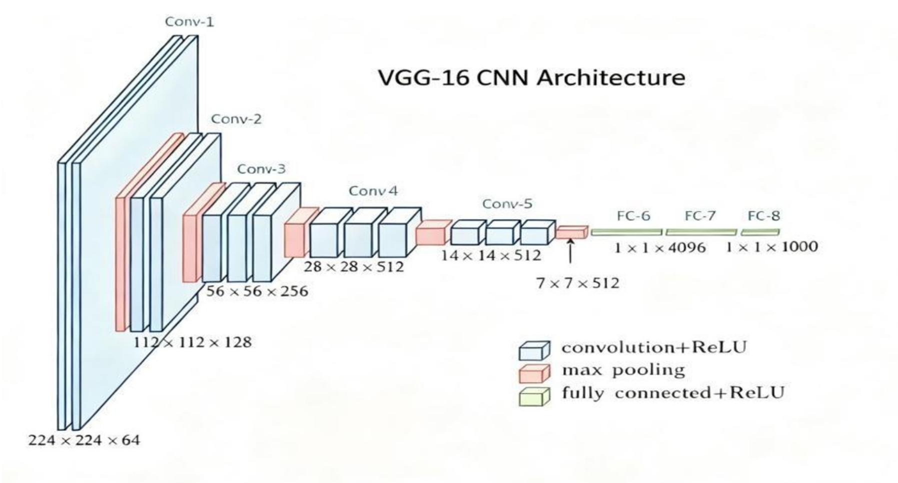

Fig. 2. CNN Architecture diagram (Picture credit: Original)

XGBoost is an ensemble learning algorithm. It builds on the "Gradient Boosting Decision Tree" (GBDT). It improves this base model through optimization. Figure 3 shows the main idea. The idea is to build many decision trees. These trees are built one after the other. Each new tree tries to fix the mistakes of the trees before it. Due to its high efficiency and strong generalization ability, it is widely used in classification, regression, and other machine learning tasks. XGBoost improves the model over many rounds. Each round uses gradient descent. The goal is to make the loss function as small as possible. This process slowly makes the model better. The algorithm can also rank features by importance. This ranking helps people understand the model's predictions. It also explains why the model makes its choices. For these reasons, XGBoost is a helpful tool for predicting short-term passenger flow[10].

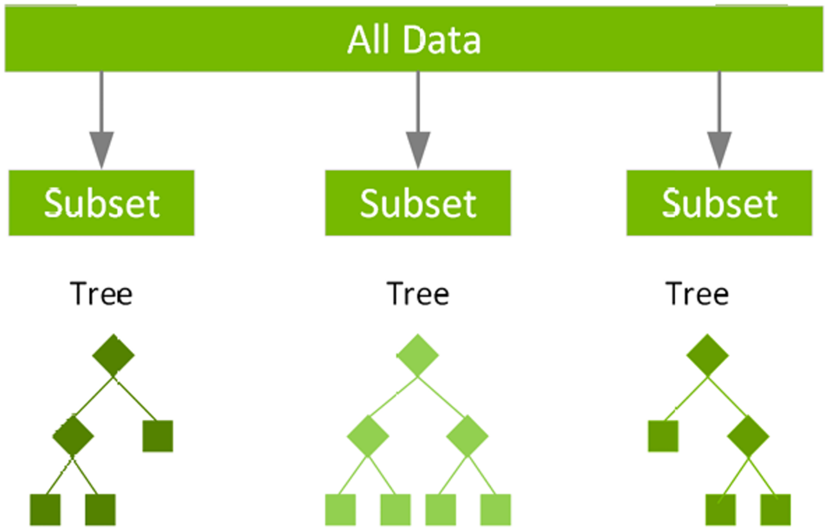

Fig. 3. XGBoost Architecture diagram [10]

STGCN is a deep learning model. It is made to work with spatiotemporal graph data. Its structure uses stacked spatiotemporal blocks. Graph convolution layers take care of space. They find how different points on a map relate to each other. Gated time convolution layers take care of time. They study the patterns in the sequence of data. A final fully connected layer gives the prediction result. This design is lightweight and uses local operations. This makes the model much faster to run. It also needs fewer parameters. This way of modeling is very good for traffic. It can catch changing traffic patterns well. It also finds the rules of how traffic changes over both space and time inside a city [11].

2.5 Introduction to Evaluation Metrics

For systematically evaluating and comparing the performance of different predictive models, this study considers both prediction accuracy and computational efficiency, selecting five commonly used evaluation metrics: MAE, RMSE, MAPE, R^2 , and training time t.

MAE is the Mean Absolute Error. It is the average of all absolute errors. Each error is the difference between a predicted value and the actual value. This metric is not easily swayed by extreme values. It gives a clear view of the average prediction error size. A smaller MAE value means the model's predictions are more accurate.

RMSE is the square root of the average squared errors. Squaring the errors penalizes large mistakes more, making RMSE sensitive to outliers. A smaller RMSE indicates a more stable and accurate model.

MAPE is the Mean Absolute Percentage Error. It is calculated as the average of all absolute relative errors. The result is shown as a percentage. This metric removes the effect of the data's own size. This removal makes comparisons easier. You can compare models on different datasets. You can also compare predictions for problems of different scales. A model with a lower MAPE value is more accurate.

R^2 is the coefficient of determination. It measures how well a model explains changes in the target data. It shows the fit between predictions and actual values. R^2 scores range from

0 to 1. A score closer to 1 means the model fits the data well. It also means the model's explanatory power is strong.

Training time refers to the amount of computational time a model takes to complete a full learning process on the training dataset, usually measured in seconds. This metric is used to evaluate the efficiency of model training. In applications that require real-time or fast responses, a shorter training time is an important advantage.

3 Method

The CNN-LSTM Hybrid Model leverages the spatial feature extraction capabilities of Convolutional Neural Networks (CNN) alongside the time series modeling strengths of Long Short-Term Memory networks (LSTM), making it adept at capturing spatiotemporal patterns in passenger flow data. In this study, the convolutional and pooling layers of the CNN model are employed to extract features from sample data, transforming them into abstract information features. Subsequently, this data is processed through a Flattening layer and then fed into the LSTM model for correlation prediction, with the final prediction output generated through a fully connected layer. Utilizing CNN for feature extraction and abstraction of sample data reduces complexity and dimensionality, thereby preventing overfitting due to data anomalies. The LSTM effectively filters and updates relevant information in time-series data, enhancing the prediction accuracy and generalization ability of the model. The structure of the CNN-LSTM network is shown in Figure 4.

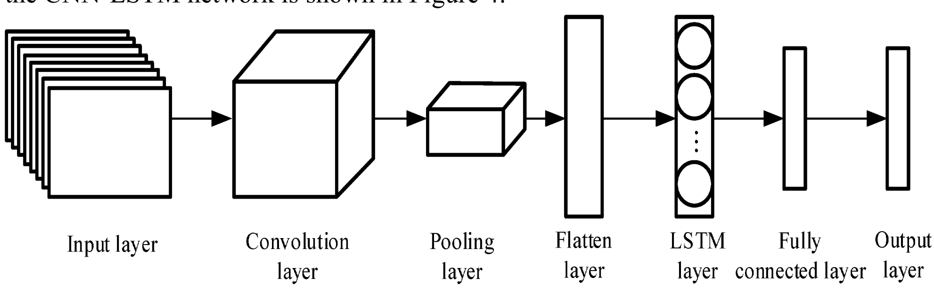

Fig. 4. CNN-LSTM Network Structure [6]

The aim of employing XGBoost for predicting passenger flow is to identify a suitable model and continually refine the parameters to reduce the discrepancy between predicted and actual values. Consequently, this article outlines the objective function of the XGBoost model and seeks to minimize it in order to determine the optimal parameters. In this article, the objective function of the XGBoost model is represented as:

$$obj = \sum_{i=1}^{n} l\,(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{K} \Omega(f_k) \tag{1}$$

The objective function consists of two parts. The first part is the loss function, which measures how well the model fits the training set. The smaller the value of the loss function, the better the fit. The second part is the regularization term, which measures the complexity of the model. Optimizing the regularization term can help prevent poor generalization.Regularization term $\Omega(f_k)$ is expressed as:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} \omega_j^2 \tag{2}$$

The model has T leaf nodes. Each leaf node gets a score. The paper calls the score for the j-th leaf ω_j. The term γT measures the tree's complexity based on the number of leaves. The parameter λ is a coefficient. This coefficient helps control the total number of leaves. The XGBoost model runs for many iterations. The paper looks at the objective function after the t-th iteration.

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t)}) + \sum_{k=1}^{t} \Omega(f_k) = \sum_{i=1}^{n} l(y_i + \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) + constant \quad (3)$$

The constant term comes from earlier trees. It is the regularization penalty from the first t-1 iterations.The algorithm uses a second-order Taylor expansion. This expansion approximates the error term in the objective function. After this step, the algorithm can be updated.

$$obj^{(t)} \approx \sum_{i=1}^{n} \left[l(y_i, \hat{y}_i^{(t)}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t) + constant \quad (4)$$

The term g_i represents the first derivative of the error function. The term h_i represents the second derivative of the error function. Their expressions are shown below.

$$g_i = \delta_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}), \; h_i = \delta^2_{\hat{y}_i^{(t-1)}} (y_i, \hat{y}_i^{(t-1)}). \quad (5)$$

STGCN is applied to forecast subway passenger flow. Through the process shown in Figure 5, the prediction of subway passenger flow is achieved. The overall process starts by inputting the historical time-step node feature sequences(v_{t-M+1} , ... v_t) from the left region. Here, M is the length of the time window, and v represents the features of each node. Then, multiple ST-Conv Blocks (spatiotemporal convolution blocks) sequentially extract spatiotemporal features from the input. Each layer passes features through weight sharing (W), gradually learning deep spatiotemporal dependencies. Finally, through the Output Layer, the predicted values $\hat{v}$ for future time steps are output. Additionally, from the intermediate upper-level temporal gated convolution, a temporal gated convolution is applied to the feature layer v^l of the input along the time dimension to capture dynamic patterns in the time series, producing features with 64 output channels. Then, based on the graph's adjacency relationships, a spatial graph convolution is applied to the features after temporal convolution to capture spatial dependencies between nodes (such as the traffic influence between adjacent road segments), outputting features with 16 channels. Finally, a temporal gated convolution is applied again to the spatially convolved features to integrate spatiotemporal features, producing the features for the next layer v^{l+1}.Finally, the rightmost part performs a 1-dimensional convolution on the input time series features (v^l_{t-M+1}, ... v^l_t) to capture local patterns along the temporal dimension. The output of the 1D convolution is then fed into a Gated Linear Unit (GLU), which controls the flow of features through a gating mechanism (similar to a "switch," allowing the model to focus on important temporal patterns), ultimately producing the processed time series ($v^l_{t-M+K_t}$, ... v^l_t)where Kt is a parameter related to the kernel size of the temporal convolution. Through these steps, STGCN captures dynamic dependencies along the temporal dimension and graph structural dependencies along the spatial dimension, ultimately completing the short-term subway prediction task for spatiotemporal sequences.

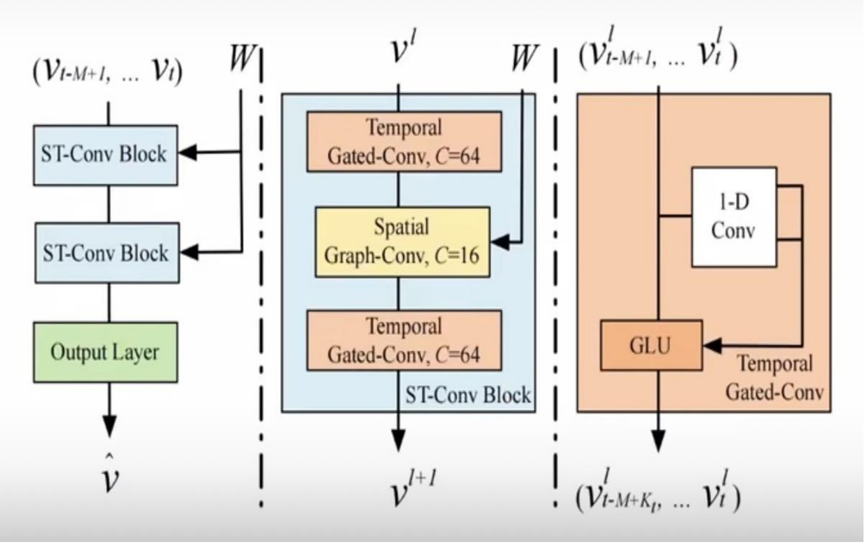

Fig. 5. STGCN Passenger Flow Forecasting Flowchart (Picture credit: Original)

4 Results

Figure 6 shows the final prediction results. These results are for short-term subway passenger flow. The models used are CNN-LSTM, XGBoost, and STGCN.

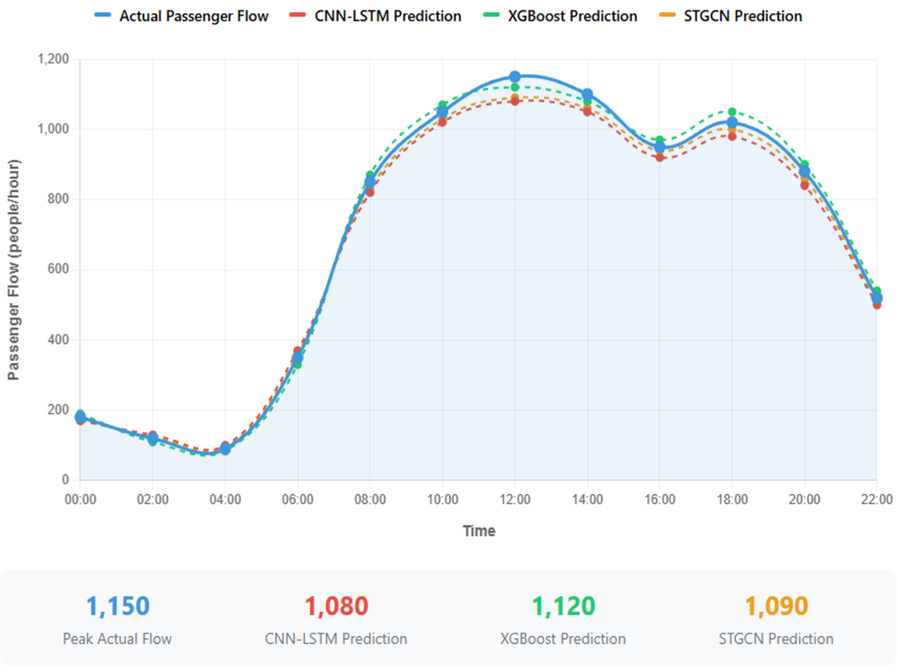

Fig. 6. Comparison of 24-hour Passenger Flow Forecasts at Subway Stations (Picture credit: Original)

Next, the three models were evaluated. The models are CNN-LSTM, XGBoost, and STGCN. The evaluation used several metrics. These metrics are MAE, RMSE, MAPE, R^2, and training time in seconds. The results of this evaluation are presented in Table 2.

Table 2. Result charts of CNN-LSTM, XGBoost, STGCN based on five evaluation metrics

Evaluation metrics	CNN-LSTM	XGBoost	STGCN
MAE	23.4	28.7	22.9
RMSE	35.6	41.2	33.8
MAPE	8.7%	10.3%	7.9%
R^2	0.923	0.897	0.941
Training time	45	12	68

The experiment shows a clear result. The three models do not perform the same. Their performance differs in short-term passenger flow prediction. As shown in Table 2, STGCN is the most accurate in capturing the morning and evening peak trends, with fluctuations that closely match the real curves; CNN-LSTM performs well in trend fitting but shows slight lag at the peaks; while XGBoost, although consistent with the overall trend, responds more slowly to sudden changes and has weaker ability to capture details.

Table 2 gives detailed performance numbers. STGCN has the best scores for MAE, RMSE, and MAPE. It also has the highest R^2 value. This shows its strong ability to model both space and time. CNN-LSTM is slightly less accurate than STGCN. However, it is more accurate than XGBoost. XGBoost trains much faster than the other models. This speed makes it good for tasks that need quick results. Its prediction accuracy is lower than the other two models.

5 Conclusion

This study addresses short-term passenger flow forecasting for city subways. It has completed a series of related tasks. First, based on historical metro passenger flow and weather data in Shenzhen, the differential impacts of various weather factors (rainfall, wind, high temperature) on weekday and weekend passenger flow were quantitatively analyzed, confirming that meteorological information is an important factor for enhancing the robustness of forecasting models. Second, this paper systematically constructs and implements three representative forecasting models: CNN-LSTM, a deep learning hybrid model capable of capturing complex spatiotemporal dependencies; XGBoost, a machine learning model known for strong feature engineering and high efficiency; and STGCN, an advanced deep learning model specifically designed for spatiotemporal graph data. Finally, through rigorously designed experiments, a comprehensive performance evaluation and comparison of the above models were conducted from five aspects: MAE, RMSE, MAPE, R^2, and training time.The excellent performance of STGCN is attributed to its joint modeling of spatiotemporal dependencies. Graph convolution effectively captures the interactions of passenger flow between stations, while temporal convolution enhances the ability to capture periodic and trend changes. Although CNN-LSTM has the potential for spatiotemporal processing, its structure lacks explicit modeling of spatial relationships, making it slightly less effective in complex road networks. As a non-temporal model, XGBoost struggles to effectively handle dynamic dependencies in passenger flow sequences, limiting its prediction accuracy.

Future work could explore two paths. The first is to find more lightweight spatiotemporal network designs. The second is to combine STGCN and XGBoost into one model. The goal is to make the model much faster to train and run. This must happen without losing its high accuracy. Such improvements would meet the needs of online, real-time prediction. Incorporating more external factors such as real-time weather information, emergency public events, social media data, and surrounding points of interest (POI) into the model inputs could further enhance the model's robustness and accuracy in predicting unusual passenger flow (such as during holidays or large events). The STGCN in this study relies on a predefined static graph structure of the subway network. Future work could explore dynamic graph neural networks, enabling the model to adaptively learn the dynamic, implicit spatiotemporal correlations between stations based on real-time passenger flow, thereby more precisely characterizing the propagation and evolution of passenger flow.

References

1. D. Q. Liu, Z Wu, S. R. Sun, Study on Subway passenger flow prediction based on deep recurrent neural network, Springer Science+Business Media, LLC, part of Springer Nature (2020)
2. P. Zhao, L. Li, Prediction of urban Rail Transit station Inflows Based on ARIMA Model, Journal of Chongqing Jiaotong University (Natural Science), 39, (2020), http://doi.org/10.3969/j.issn.1674-0969.2020.01.07
3. M. Z. Zhao, D. Wang, J. Fang,Y. Li, J. Mao, Prediction of subway station temperature based on LSTM neural network, Journal of Beijing Jiaotong University, 44, (2020), http://doi.org/10.11860/j.issn.1673-0291.20190136
4. Y. X. Sun, B. Leng, W. Guan, A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system, www.elsevier.com/locate/neucom,(2015)
5. J. J. Shi, P. Yang,Y. N. Pi, Machine Learning-based Simulation Research of On-line Subway Pedestrian Flow Control, Journal of System Simulation, 35, (2023), http://doi.org/10.16182/j.issn100473x.joss.21-0835
6. J. Liu, Q. L. He,Z. K. Yue, Y. L. Pei, A Hybrid Strategy-Improved SSA-CNN-LSTM Model for Metro Passenger Flow Forecasting, Mathematics, 12, 3929, (2024), http://doi.org/10.3390/math122443929
7. X. Q. Long, J. Li, Y. R. Chen, Metro short-term traffic flow prediction with deep learning, Control and Decision, 34, (2019), http://doi.org/10.13195/j.kzyjc.2018.1393
8. X. L. Han, Y. Luo, Impacts of weather Factors on subway passenger Flow of Beijing, Journal of Chongqing Jiaotong University (Natural Science), 39, (2020), http://doi.org/10.3969/j.issn.1674-0696.2020.10.01
9. Y. J. Park,Y. H. Choi,K. Kim, K. Y. Jae, Machine learning approach for study on subway passenger flow, natureportfolio, (2020), http://doi.org/10.1038/s41598-022-06767-7
10. M. H. Yao,G. Gao, Short-Term Passenger Flow Prediction of Subway Based on XGBoost Multi-feature Fusion, ACM ISBN, 979-8-4007-0766-7/23/09, (2023), http://doi.org/10.11.45/3627377.3627393
11. X. S. Wang,W. Bai,Z. K. Meng,B. B. Xin,R. F. Gao,X. J. Lv, The Prediction of Flow in Railway Station Based on RCC-STGCN, Digital Object Identifier, (2023), 10.1109/ACCESS.2023.3334280