# Short-term Passenger Flow Prediction of Metro based on ARIMA and LSTM Models

Wushuang Fan *

School of Economics and Management, East China Normal University, Shanghai, 200000, China

* Corresponding Author Email: 10224804415@stu.ecnu.edu.cn

**Abstract.** The essay studies the comparative effectiveness of ARIMA and LSTM models in predicting passenger loads in metro systems, emphasizing the Beijing metro as a case study. It has been argued that accuracy is essential when it comes to forecasting passenger flows, which will, in turn, enhance the efficiency of urban mass transit systems and promote a comfortable and preferable traveling experience for the passengers. The article meticulously dissects and compares the techniques proposed in this paper in a detailed fashion by revealing each of their strengths and weaknesses and historical passenger load numbers. The techniques are illustrated using data from the AFC system of the Beijing Metro, focusing particularly on Wangfujing Station for its tremendously high passenger flow. The effectiveness of these techniques is shown on a variety of evaluation metrics, including mean squared error (MSE), mean absolute error (MAE), root mean squared error (RMSE), and $R^2$. It is found that the ARIMA model is better suited to capture shorter intervals (2 minutes) because of its lower error rates and that it leads to greater clarity in longer intervals (5 minutes) despite having higher error rates. This research contributes to the broader field of intelligent transportation systems by providing insights into the effective application of predictive modeling techniques for enhancing metro system operations.

**Keywords:** Metro; ARIMA; LSTM; passenger flow prediction.

## 1. Introduction

Urban rail transit is now a crucial choice for residents to commute in an environmentally friendly way in today's society. However, issues have arisen due to urban expansion and population growth. Operators must utilize efficient scientific techniques to manage increases in passenger volume during peak morning and evening hours, severe weather conditions, and emergencies. The crucial aspect is to forecast passenger movements precisely soon. Predicting accurately is formidable because various elements influence passenger flow, including station location, building size, adjacent traffic, weather, and emergencies [1].

Numerous domestic and foreign academics have extensively explored short-term passenger flow forecasting methodologies, with a predominant reliance on time series analytic approaches. The studies can be classified into two primary categories according to the data source: univariate forecasting depends on alterations in passenger flow data, whereas multivariate forecasting integrates external elements such as weather conditions, climate change, and holidays to forecast passenger flow [2].

In the past, research on univariate prediction focused on individual predictions mainly for mathematical statistics and machine learning methods. For instance, as pointed out by Gung et al. and their team, they applied fuzzy C-means clustering and linear regression to model passenger flow in stations.; they tried and came up with what appeared to be a fairly good performance [3]. The average error rate for all the stations is 0.27%, and for an individual station, it is 3.92%. Due to its robust computing capabilities on time series data, transportation analysts often adopt the ARIMA model. Wu and his colleagues employed the ARIMA framework to estimate the passenger quantity in particular stations from fall to early winter 2019 [4]. They demonstrated the proficiency of the model in anticipating the evolution of passenger quantity for urban mass transit systems while keeping the error rate under 10 percent.

Greater utilization of deep learning has facilitated the expansion of research in the domain of predicting passenger volume on rail transport. The LS-GCN model, devised by Han, incorporated the

extended short-term memory network and graph convolutional network, to yield accurate traffic flow forecasts by offering high precision in prediction. The LS-GCN model efficiently seized the spatiotemporal features of traffic information and truncated the number of factors. Hence, it prudently advanced the preciseness level of the traffic forecast [5]. Furthermore, Chang's ARIMA-LSTM model integrated the linear forecasting capacity of ARIMA with the computational prowess of LSTM. It was designed to process the data of an ARIMA residual series and displayed more efficient results when compared to strictly ARIMA or LSTM models [6].

In multivariate passenger flow forecasting studies, the analysis usually focuses on the influence of external factors such as weather and holidays. Zhang extracted vital factors such as operating hours and air temperature through regression analysis and integrated them into the LSTM model, thus improving the interpretability of the model [7]. Li et al. identified significant weather and historical data factors using Pearson analysis, which verified the high accuracy of LSTM in short-term passenger flow prediction [8]. Liu et al. developed an ARIMA-GARCH model by integrating an ARIMA model with a Generalised AutoRegressive Conditional Heteroskedasticity (GARCH) model. They concentrate on predicting vacation passenger traffic by efficiently capturing its features and expediting the prediction process [9]. Zhang et al. used the graph convolution technique to combine spatiotemporal and external factors (e.g., weather and holiday information) to improve the accuracy of transit data prediction [10].

Although there have been numerous studies on metro passenger flow prediction in recent years, and many of them have increased the depth and breadth of their research by comparing deep learning algorithms with the ARIMA algorithm, there are relatively few detailed comparative analyses between the ARIMA algorithm and the LSTM algorithm. Therefore, this study aims to focus on exploring and comparing the performance of the ARIMA algorithm, which is popular among traditional prediction algorithms, with the LSTM algorithm, which is widely used in the field of deep learning, based on a univariate dataset, in terms of effectiveness and prediction accuracy.

## 2. Methods

### 2.1. Data Source

The data in this study were obtained from the Beijing metro's Automatic Fair Collection (AFC) system. The data were collected from the underground entry and exit records in May 2019 at all stations in Beijing. As shown in Table 1, the data are stored in CSV files, including the IC card number, departure time, and terminal way.

**Table 1.** Attribute information for raw data

| Field | Instruction | Data Type |
|---|---|---|
| IC_card_No | IC card number | Object |
| Started_line | Starting Line | Object |
| Started_at | Departure time | Object |
| Ended_at | Arrival time | Object |
| Ended_No | Terminal number | Object |
| Ended_line | Final destination line | Object |
| Started_No | Starting station number | Object |
| Started_Name | Starting Station Name | Object |
| Ended_Name | Destination name | Object |

To highlight the main research questions, the researcher extracted the name of the starting station used for the study and the time of entry deleted unnecessary columns, and statistically processed the data from one bar to merge it into meaningful research objects for further analysis.

### 2.2. Indicator Selection and Description

In this study, the researchers took 2min and 5min as the minimum time granularity, respectively, and selected Wangfujing station with more passenger flow as a specific station for passenger flow inbound analysis, counted the number of people progressing in Wangfujing in each time granularity within three days of working days, and used this as an index to measure the short-time passenger flow of the station. Within a short time granularity, the inbound passenger flow reflects the current actual passenger flow situation and predicts the possible future trend. This can guide for optimizing operational efficiency and making timely arrangements.

### 2.3. Method Introduction

#### 2.3.1. ARIMA model

$ARIMA(p,d,q)$ model is known as the differential autoregressive moving average model. $AR$ is autoregressive, $p$ is the autoregressive term. $MA$is the moving average, $q$ is the number of moving average terms; $d$ is the number of differentials made when the time series becomes smooth. The $ARIMA$ model aims to convert the original non-smooth series into a soft series by differential transformation of the time series. Then, on this smooth series, the model forecasts the current values as a linear combination of the values of the past number of periods and the error term. The following is the general formula for the ARIMA model:

$$\left(1-\sum_{i=1}^{p}\varphi_i L^i\right)(1-L)^d X_t = \left(1+\sum_{j=1}^{q}\theta_j L^j\right)\varepsilon_t \tag{1}$$

The $(1-L)^d X_t$ section represents the difference component, the initial stage in model development. The determination of the difference order d relies on the Augmented Dickey-Fuller (ADF) test outcomes for stationarity. The ADF test is a frequently employed test for assessing the smoothness of a time series by testing for the presence of a unit root, indicating non-smoothness. Suppose the test statistic is smaller than the crucial value or if the $p-value$ is less than the significance threshold. The first premise is disproved in this scenario, and the time series is smooth. Differencing removes seasonality or trends in non-stationary data using first-order or higher-order differencing. This ensures that the statistical properties of the data, such as mean and variance, remain constant across time, preparing the data for $ARMA$ modeling.

The calculation of the ADF statistic is based on the following regression model:

$$\Delta \mathrm{X_t} = \alpha + \beta \mathrm{t} + \gamma \mathrm{X_{t-1}} + \sum_{\mathrm{i=1}}^{\mathrm{p-1}} \delta_{\mathrm{i}}\, \Delta \mathrm{X_{t-1}} + \varepsilon_{\mathrm{t}} \tag{2}$$

The formulas for the first and second-order differences are as follows, and so on for higher-order differences:

$$\Delta y_t = y_t - y_{t-1} \tag{3}$$

$$\Delta^2 y_t = \Delta y_t - \Delta y_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) = y_t - 2y_{t-1} + y_{t-2} \tag{4}$$

After determining the difference order d of the data, they were followed by the $ARMA$ part, i.e., determining the values of p and q. Here, the researcher used the Statsmodels library for Python to automate the implementation; the methodology adopted was to find the parameters and evaluate the model according to the Akaike Information Criterion (AIC) grid.

$ARMA$ can be further divided into the processes of $AR$ and $MA$, where $AR$ refers to:

$$X_t = \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t \tag{5}$$

$MA$ refers to:

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} \tag{6}$$

The formula for judging the $AIC$ is:

$$AIC = 2k - 2\ln(L) \tag{7}$$

### 2.3.2. LSTM model

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) for processing long-time sequences. LSTM adds the filtering of past states based on RNN so that more influential states can be effectively selected. In addition, LSTM can avoid the gradient vanishing and explosion problems by extracting the long-term dependency information from the long sequence data. The LSTM network structure, as shown in Figure 1 by Visio.

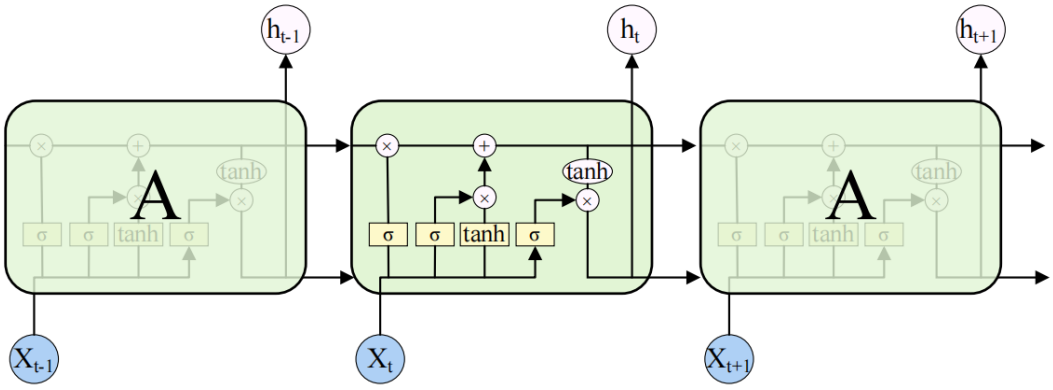


**Fig. 1** LSTM network topology [11]

Long Short-Term Memory (LSTM) is a Recurrent Neural Network (RNN) for processing long-time sequences. LSTM adds the filtering of past states based on RNN so that more influential states can be effectively selected. In addition, LSTM can avoid the gradient vanishing and explosion problems by extracting the long-term dependency information from the long sequence data. The LSTM network structure, as shown in Figure 1 by Visio.

The forgetting gate's role is to selectively filter and discard information in the cell state using output module c from the preceding cell module. The system must determine which components to retain and discard. The formulae are as follows:

$$f_t = \sigma(W_f \times [x_t, h_t] + b_f) \tag{8}$$

$$C_t = i_t \times \widehat{C}_t + f_t \times C_{t-1} \tag{9}$$

The input gate function selectively encodes and determines which new information to store in the delicate cell states (cell modules). The input gate comprises a Sigmoid layer for updating values and a Tanh layer for creating new candidate memories and incorporating them with the memories that are not retained. The Tanh layer produces fresh candidate memories for the deleted attribute data. The formula is as follows:

$$i_t = \sigma(W_i \times [x_t, h_{t-1}] + b_i) \tag{10}$$

$$\widehat{C}_t = \tanh(W_c \times [x_t, h_{t-1}] + b_i) \tag{11}$$

The input gate result, denoted as $\widehat{C}_t$, is determined by the $i_t$ value to decide whether $\widehat{C}_t$, is included in the state at moment $i$. The two values are ultimately multiplied to get the final output information. The formula is as follows:

$$O_t = \sigma(W_o \times [x_t, h_{t-1}] + b_o) \tag{12}$$

$$h_t = O_t \times \tanh(C_t) \tag{13}$$

Where $h_t$ is the final output of the output gate.

### 2.3.3. Model evaluation indicators

Model evaluation involves reviewing one or more existing models and evaluating their performance based on their category using various methods. Model assessment involves assessing established models based on their categories, utilizing various indicators to evaluate their

performance strengths and weaknesses. There are four primary model assessment techniques: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared ($R^2$). The formulas are as follows:

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|(y_i - \hat{y}_i)| \tag{14}$$

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \tag{15}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{16}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \tag{17}$$

## 3. Results and Discussion

### 3.1. ARIMA Model Results

This study uses the matplotlib library in Python to visualize the original dataset, shown in figure 2. Horizontal coordinates indicate hours, and vertical coordinates indicate the number of persons (two-minute intervals).

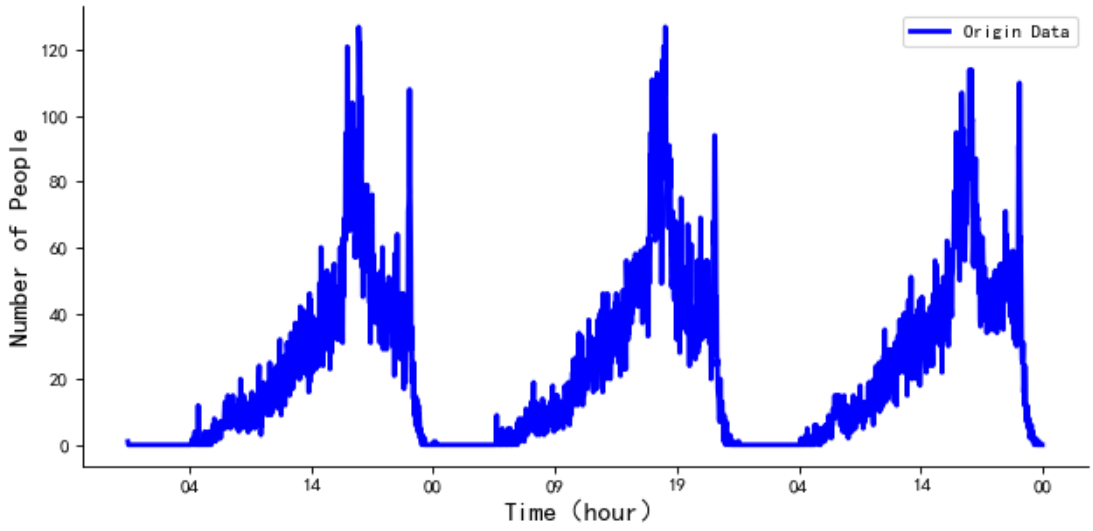


**Fig. 2** Original data

Researchers then performed the ADF test on the original data (all following are in two-minute intervals). As shown in Table 2, the ADF value of these time series data is -2.555, and the p-value of the ADF test is 0.102. The critical values for 1%, 5%, and 10% are -3.438, -2.864, and -2.568, respectively. Since the p-value of 0.102 is more significant than 0.1, the null hypothesis cannot be rejected, and the series is not stationary. This indicates the need for a new ADF test and first-order differentiation of the series.

**Table 2.** ADF test

| Differencing Order | ADF | P | Critical Value | | |
|---|---|---|---|---|---|
| | | | 1% | 5% | 10% |
| 0 | -2.555 | 0.102 | -3.438 | -2.864 | -2.568 |
| 1 | -7.208 | 0 | -3.438 | -2.864 | -2.568 |

The first-order difference of the data shows that the null hypothesis is rejected. The confidence level exceeds 99%, resulting in a p-value of less than 0.01. This time, the sequence is stationary. Clearly, d should be equal to 1.

The values of p and q were then determined based on the AIC. Overall, the lower the AIC value, the more favored the model. Table 3 shows the part of AIC values for different p and q values. The author looked for the model with the smallest AIC value, which had the best balance between fitting the data and maintaining parsimony.

**Table 3.** AIC test

| (p, d, q) | AIC |
|---|---|
| (0, 1, 0) | 7501.470 |
| (0, 1, 1) | 7492.435 |
| (0, 1, 2) | 7465.324 |
| (0, 1, 3) | 7434.066 |
| (1, 1, 0) | 7495.167 |
| (2, 1, 2) | 7426.775 |
| (2, 1, 3) | 7400.335 |
| (2, 1, 4) | 7401.942 |

The optimal p, d, and q values were determined to be (2, 1, 3) after automatic parameter finding based on AIC. Then, 80% of the data was taken as the training set for model training and prediction. After this, model testing is also done with the help of residual calibration. With the help of a standard probability plot, figure 3, also known as the Q-Q (Quantile-Quantile) plot, is used to assess whether the distribution of residuals is approximately normal. There is also a histogram of the distribution of the residuals in figure 3, which ideally should show a bell curve if the data are typically distributed.

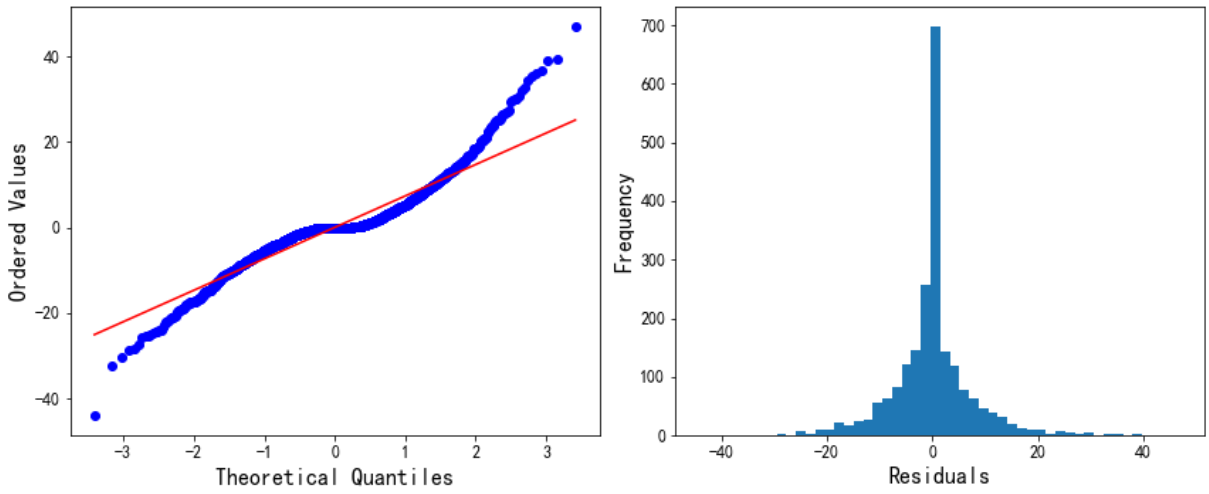


**Fig. 3** Quantile-Quantile plot and Histogram

Combining these two graphs, this paper can conclude that the data roughly conforms to a normal distribution in the middle range. However, there may be some deviation at the ends, particularly in the left tail. Then, the calculated evaluation metrics are shown in table 4, and the predicted results are shown in figure 4.

**Table 4.** Indicators for evaluation

| MSE | MAE | RMSE | $R^2$ |
|---|---|---|---|
| 89.407 | 6.961 | 9.456 | 0.842 |

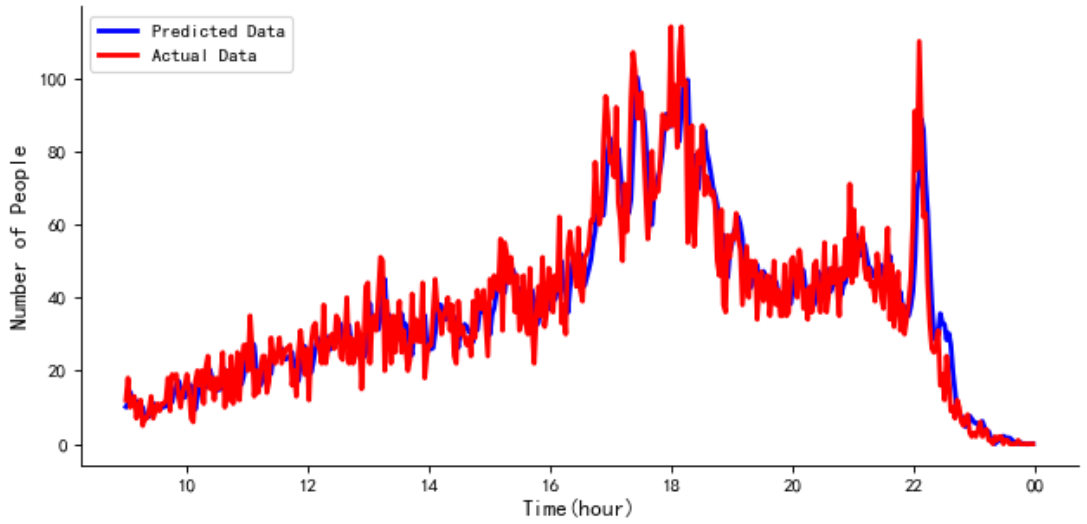


**Fig. 4** Predicted Results (ARIMA; 2min interval)

Combining the above metrics and the graph's predicted vs. actual data lines (blue and red curves), this paper can conclude that the model's overall fit is quite good because of the high $R^2$ value. The charts show that the predicted and actual data trends are generally in line, but slight differences exist, especially in the peaks. There is a general agreement regarding the overall trend, but specific fluctuations are missing. The results of training the dataset at five-minute intervals are shown in figure 5.

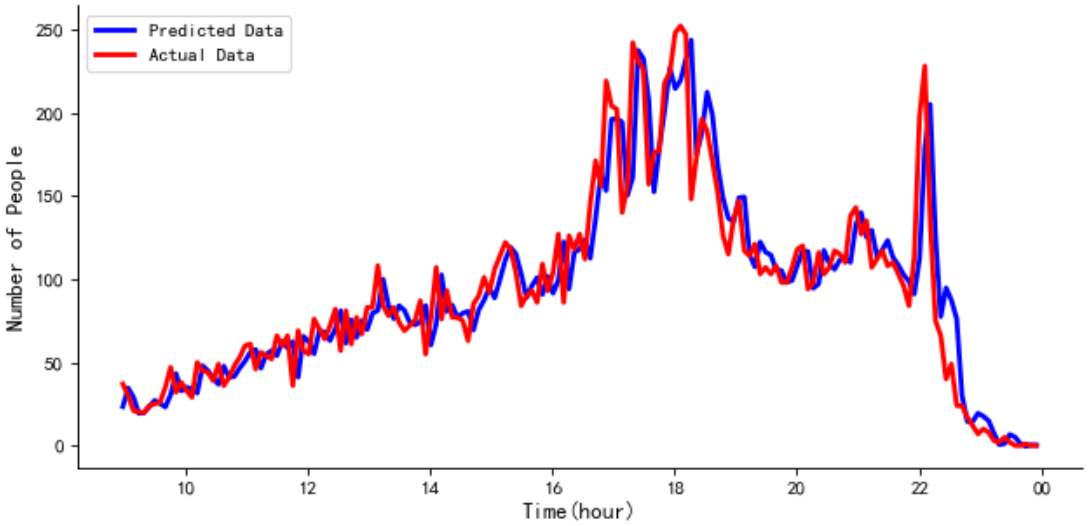


**Fig. 5** Predicted Results (ARIMA; 5min interval)

Charts with two-minute intervals may show more detail because the data points are denser. This may help reveal fluctuations or patterns over a shorter period. Charts with five-minute intervals may be smoother because each point represents an average or aggregated value over a longer time frame. This may help to see trends over a more extended period. In terms of differences in predictive accuracy, in the two-minute interval charts, the gap between predicted and actual values may be more pronounced at some points, possibly because more frequent data points reveal limitations in the predictive model. In charts with five-minute intervals, the difference between predicted and actual

data may be less obvious, possibly because the smoothing effect of the data reduces the noise, making the predictions appear more accurate.

### 3.2. LSTM Model Results

The LSTM model utilizes a Sequential design to allow for the incremental development of network layers. The initial layer of the model consists of an LSTM layer set up with 128 neurons (units). The layer utilizes the Rectified Linear Unit (ReLU) activation function to improve the model's nonlinear representation. A Dropout layer with a dropout rate 0.2 is put after the LSTM layer. This layer aims to mitigate model overfitting and enhance the model's generalization by randomly discarding a fraction of the neurons' outputs. The network's last layer is the fully connected layer (Dense), with a single neuron responsible for producing the final prediction of the model. This layer has a default linear activation function and is appropriate for regression problems. The model employs Mean Squared Error (MSE) as the loss function and utilizes the Adam optimizer for parameter optimization. The model undergoes 100 training sessions (epochs) with a batch size of 128 samples processed in each batch. Figure 6 is the dataset for two-minute intervals and figure 7 is the dataset for five-minute intervals.

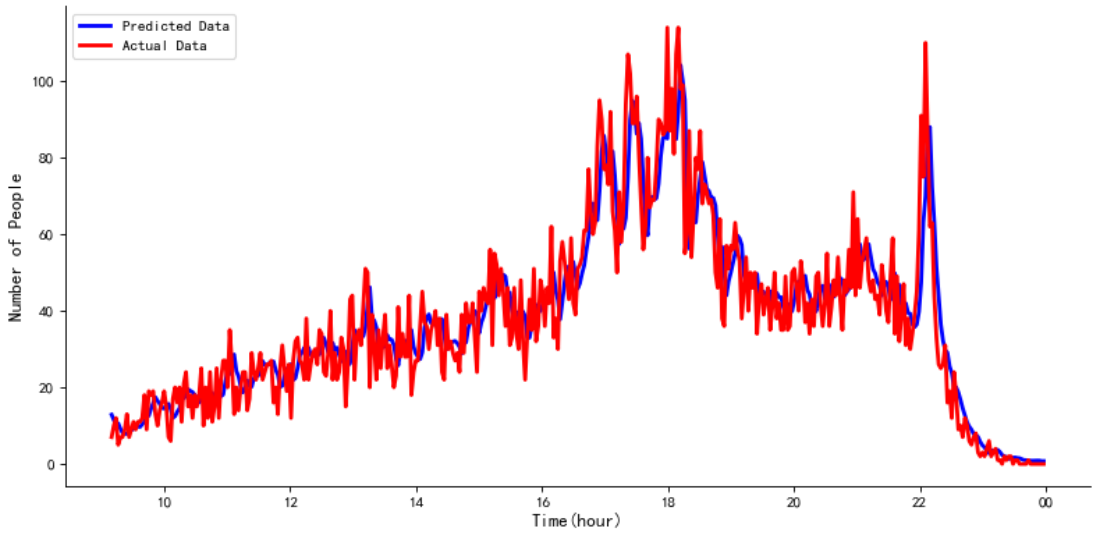


**Fig. 6** Predicted Results (LSTM; 2min interval)

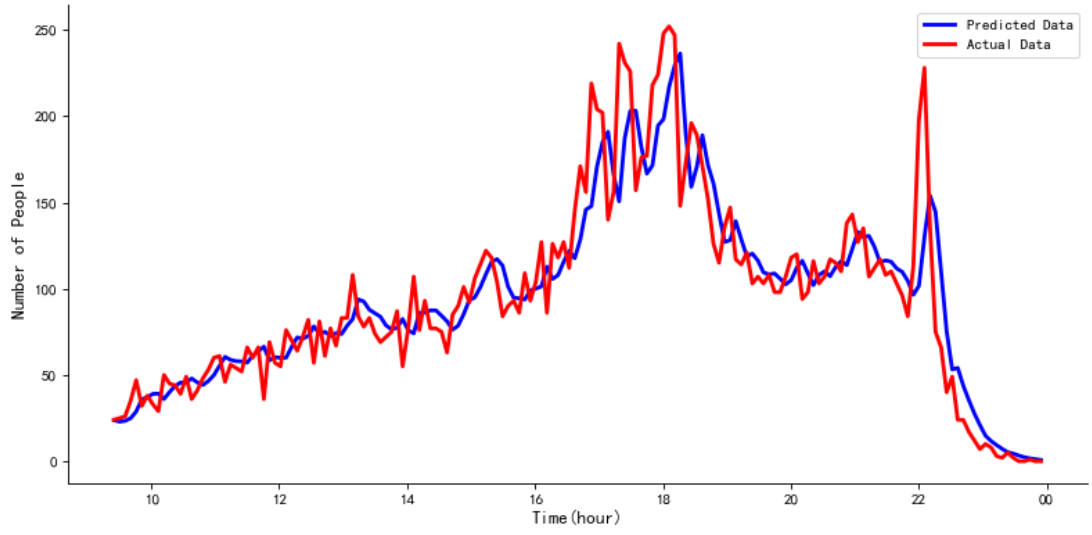


**Fig. 7** Predicted Results (LSTM; 5min interval)

Regarding predictive accuracy, the predicted curve (blue) closely follows the actual curve (red) in the graphs for both time intervals. This suggests that the model can capture the main trends in the

data regardless of whether it is a 2-minute or 5-minute interval. The ARIMA model fits the data better in the 5-minute interval image of the LSTM compared to the 5-minute interval image of the ARIMA, which suggests that intuitively, at the 5-minute interval time granularity, the ARIMA model is more suitable for univariate passenger flow prediction.

### 3.3. Discussion of Two Models

The present investigation examined a sophisticated model to resolve a particular issue. Nevertheless, the model failed to produce the anticipated outcomes, which might be ascribed to various factors. Firstly, the sample size was inadequate, which limited the model's learning ability and impacted its performance. Secondly, the model parameters were not optimized, and further adjustments are necessary to enhance the model's outcomes.

It is important to note that a complex model's performance in multivariate situations cannot be directly compared to its performance in univariate situations. Since complex models have a broader modeling space, they should ideally demonstrate superior performance in all situations. The study suggests that the inability to find the best model within a vast model space may be another factor contributing to poor performance.

To enhance the effectiveness of the model, it is recommended that future research prioritize the augmentation of the sample size and the ongoing refinement of model parameters. These enhancements will verify intricate models' capability in tackling particular issues and augment their practical efficacy.

### 3.4. Inference Analysis

Taking the indicators together, the results are as follows (Table 5):

**Table 5.** Indicators for evaluation of models

| Model | Time Granularity | MSE | MAE | RMSE | $R^2$ |
|---|---|---|---|---|---|
| ARIMA | 2 minutes | 89.407 | 6.961 | 9.456 | 0.842 |
| ARIMA | 5 minutes | 470.641 | 14.313 | 21.694 | 0.858 |
| LSTM | 2 minutes | 91.265 | 6.908 | 9.553 | 0.838 |
| LSTM | 5 minutes | 537.261 | 15.427 | 23.178 | 0.835 |

The ARIMA model has decreased MSE, MAE, and RMSE metrics when the period is 2 minutes, suggesting that it has reduced prediction error at the 2-minute time granularity. The $R^2$ values are very close to those of the LSTM model, but still slightly higher than the LSTM model, which indicates that the ARIMA model still has an excellent ability to elucidate variability compared to the LSTM model.

When time intervals were set at 5 minutes, the ARIMA model significantly outperformed the LSTM model regarding MSE, MAE, and RMSE. In addition, the ARIMA model also performs slightly better on the coefficient of determination ($R^2$), implying that it is slightly more effective in explaining data variability. This is consistent with the results of the image observations.

The usage of the ARIMA model led to a noteworthy rise in the mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE) indices as interval duration was augmented from 2 to 5 minutes. This seems to suggest that forecast error is positively correlated with period length. Nonetheless, worth noting, there was an improvement in the $R^2$ coefficient, perhaps due to the model's capability to detect trends over longer periods even when a substantial amount of error is still present.

An observation of note in the case of the LSTM model is that when the time interval was widened from 2 to 5 minutes, the forecast error increased markedly. In contrast to ARIMA, the $R^2$of the LSTM experienced a drop, which indicates that LSTM model is more responsive to variations in the time interval when capturing the overall temporal dynamics of the data.

## 4. Conclusion

This research contrasts the forecasting capabilities of two methods, ARIMA and LSTM, examined at different granularities (2-minute and 5-minute intervals). The results show that over the 2-minute horizon, the predictive capability of ARIMA is only marginally better than that of LSTM, as witnessed by overall lower error metrics (MSE, MAE, and RMSE). However, ARIMA is on average less able to capture data variability; especially, quite surprisingly, ARIMA, which although showing greater error metrics, would be more effective in explaining $R^2$values within the 5-minute horizon. If there is a need for accurate forecasts and the dataset is more linear, it may be more appropriate to utilize the ARIMA model at 2-minute intervals. Suppose there is a need for enhanced explanatory capability of the model and a certain level of tolerance for error. In that case, employing an ARIMA model with 5-minute intervals may be more suitable. If the dataset exhibits complexity (non-linearity) and the available resources are sufficient, the LSTM model may be ideal despite its higher error rate at 5-minute intervals. The results of this study indicate that when selecting a model, it is essential to consider both its predicted accuracy and capacity to explain the variability in the data, as well as the appropriate level of time granularity.

One constraint of this work pertains to the restricted scope of analysis, confined to two models and two distinct temporal granularities. Subsequent investigations could be expanded to encompass other time series models and a broader range of temporal granularities to authenticate the results. In addition, the datasets used in the study may have their specificity, and future research could test the models on different datasets to improve the generalisability of the findings.

Given these results, future research should continue to explore the effects of different temporal granularities on model performance and consider implementing model integration methods to improve the robustness of predictions. At the same time, research should consider the computational efficiency and practicality of the models to be more flexible and efficient in practical applications.

## References

[1] Zeng Cheng, Wu Jiayuan, Luo Xia. Literature review on short-term passenger flow prediction for urban rail transport. Railway Transport and Economy, 2021, 43(8): 105-111,125.

[2] Li Dekui, Du Shubo, Zhang Peng. Comparison of delayed passenger flow prediction methods for urban rail transit based on ARIMA and LSTM. Journal of Qingdao University of Technology, 2021, 135-142.

[3] Guang Zhirui. Research on holiday passenger flow prediction of urban rail transit. Traffic Engineering, 2017, 17(3): 27-35.

[4] Wu Xiangbin, Liu Zhifeng, Ding Chenglong, et al. Analysis and prediction of metro passenger flow data based on the ARIMA model. Defence Manufacturing Technology, 2021, 4: 15-17.

[5] Han Xu. Research on traffic flow prediction based on LSTM and GCN. Zhejiang Institute of Science and Technology, 2023.

[6] Chang Hao. Research on short-time passenger flow prediction in metros based on LSTM neural networks. Xijing College, 2023.

[7] Zhang Ping, Xiao Weizhou, Shen Zhengxi. Short-term OD passenger flow prediction of rail transit based on long- and short-term memory networks. Hebei Industrial Science and Technology, 2021, 351-356.

[8] Li Mei, Li Jing, Wei Zijian, et al., Short-time passenger flow prediction at metro stations based on deep learning long and short-term memory network structure. Urban Rail Transit Research, 2018, 42-46, 77.

[9] Liu Weiyuan, Ge Yuechun, Li Lei, et al. A study on holiday passenger flow prediction for Suzhou rail transit. Urban Express Rail Transit, 2021, 34(5): 66-73.

[10] Wei Zhang, Fenghua Zhu, Yuanyuan Chen, et al., Bus passenger flow prediction based on attention mechanisms and time-phased graph convolution. Pattern Recognition and Artificial Intelligence, 2021.

[11] Jiaxu Li, Meihan Ye, Dazhi Zhao. Passenger Flow Prediction and Risk Early Warning for Metro Cross-section based on Transportation Big Data. Highlights in Science, Engineering and Technology, 2023, 78: 41-50.