Hindawi
IET Circuits, Devices & Systems
Volume 2024, Article ID 5259452, 13 pages
https://doi.org/10.1049/2024/5259452

Research Article

Optimizing Metro Passenger Flow Prediction: Integrating Machine Learning and Time-Series Analysis with Multimodal Data Fusion

Li Wan, Wenzhi Cheng, and Jie Yang

College of Intelligent Manufacturing, Hunan University of Science and Engineering, Hunan 425199, China

Correspondence should be addressed to Li Wan; wanli@huse.edu.cn

Received 17 November 2023; Revised 15 March 2024; Accepted 5 April 2024; Published 26 April 2024

Academic Editor: Siew-Kei Lam

Accurate passenger flow forecasting is crucial in urban areas with growing transit demand. In this paper, we propose a method that combines advanced machine learning with rigorous time series analysis to improve prediction accuracy by integrating different datasets, providing a prescriptive example for passenger flow prediction in urban rail transit systems. The study employs advanced machine learning algorithms and proposes a novel prediction model that combines two-stage decomposition (seasonal and trend decomposition using LOESS–ensemble empirical mode decomposition (STL-EEMD)) and gated recurrent units. First, the STL decomposition algorithm is applied to break down the preprocessed data into trend terms, periodic terms, and irregular fluctuation terms. Then, the EEMD decomposition algorithm is employed to further decompose the irregular fluctuation terms, yielding multiple IMF components and residual residuals. Subsequently, the decomposed data from STL and EEMD are partitioned into training and test sets and normalized. The training set is utilized to train the model for optimal performance in predicting subway short-time passenger flow. The synthesis of these sophisticated methodologies serves to substantially enhance both the predictive precision and the broad applicability of the forecasting models. The efficacy of the proposed approach is rigorously evaluated through its application to empirical metro passenger flow datasets from diverse urban centers, demonstrating marked superiority in predictive performance over traditional forecasting methods. The insights gleaned from this study bear significant ramifications for the strategic planning and administration of public transportation infrastructures, potentially leading to more strategic resource allocation and an enhanced commuter experience.

1. Introduction

Urban rail transit, as an emergent modality in transportation, garners widespread public acclaim for its convenience, comfort, and environmentally friendly attributes, progressively becoming the preferred option for daily commutes and travel [1]. The influx of passengers during peak hours, particularly in the mornings and evenings, imposes considerable strain on line operations, manifesting in challenges such as station congestion and train delays [2]. In this context, the accurate prediction of passenger flow, especially through the application of multimedia data mining technologies to apprehend rapid shifts in passenger numbers, is crucial for ensuring transport safety and enhancing operational efficiency [3].

Passenger flow predictions are categorized into long-term, short-term, and short-time forecasts [4]. Long-term forecasts, relevant during the planning and construction phase of the rail network, and short-term forecasts, predicting passenger flow for the upcoming year, offer limited utility for daily operations. Conversely, short-time forecasting, which projects passenger numbers for the forthcoming 15 min, is instrumental for operational planning and train scheduling [5]. The inherent nonlinearity, nonstationarity, and randomness of short-time passenger flow complicate accurate forecasting, and the efficacy of singular models in this context is limited [6]. Notwithstanding, scholarly research into nonlinear and combinatorial optimization models has underscored the viability of predicting short-time passenger flow [7, 8]. However, traditional parametric and nonparametric models are impeded by protracted training durations and low responsiveness, rendering them ill-suited for managing large-scale data in practical applications [9]. Moreover, the extant literature on short-time passenger flow prediction predominantly concentrates on optimizing model structures and training algorithms, often overlooking the impact of multimedia data noise on prediction accuracy [10].

To address these challenges, this study introduces a novel model that integrates a two-stage decomposition approach, STL-EEMD (seasonal and trend decomposition using LOESS–ensemble empirical mode decomposition), with an enhanced gated recurrent neural network (IGRU) to refine the accuracy of short-time subway passenger flow predictions. Initially, the model employs a graph-based depth-first search algorithm to analyze passenger travel patterns within multimedia data, thereby constructing a short-time passenger flow time series. Subsequently, given the pronounced randomness and high nonstationarity of the short-time subway passenger flow sequence, the STL-EEMD method is applied to mitigate noise interference within the time series. Ultimately, an IGRU network model predicated on the residuals of gated cyclic units is formulated to facilitate precise short-time passenger flow predictions. Empirical validation utilizing multimedia data substantiates the model's efficacy in enhancing the accuracy of short-time metro passenger flow forecasts, thereby offering theoretical support for metro operators in the formulation of advanced operational strategies.

2. State of the Art

The prediction of short-time passenger flow is pivotal for the effective scheduling of train operations and passenger flow management, ensuring the timely and safe arrival of passengers at their destinations. Subway passenger flow data, though exhibiting periodic characteristics that are amenable to prediction, presents challenges due to its nonlinearity, pronounced randomness, and nonsmooth nature. Extensive research has been conducted in this domain, with forecasting methods broadly categorized into three paradigms: those based on mathematical statistics, intelligent algorithms, and hybrid models [11].

Methods grounded in mathematical statistics include commonly used models such as the Kalman filter [12], differential integrated moving average autoregressive (ARIMA) model [13], and seasonal differential autoregressive sliding average (SARIMA) model [14]. While these models are straightforward and user-friendly, their efficacy in predicting nonlinear passenger flow remains limited [15]. On the other hand, intelligent algorithm-based methods leverage their strong learning and adaptive capabilities to effectively capture the nonlinear attributes of multimedia passenger flow data. These include traditional neural networks and advanced deep learning methods such as support vector machines [16] and artificial neural networks [17]. Traditional neural networks, with their shallow structures, often fail to encapsulate complex nonlinear relationships in data, leading to significant prediction errors [18]. In contrast, deep-structure-based methods like long- and short-term memory (LSTM) and gated recurrent neural networks (GRU) have gained prominence due to their enhanced ability to capture spatio-temporal relationships. Ma et al. [19] applied LSTM networks to urban traffic flow prediction, though the complexity of LSTM's parameter determination remains a challenge. Conversely, Dai et al. [20] employed spatiotemporal analysis in conjunction with GRUs for short-term traffic flow prediction, noting that while GRUs, with their reduced gate structure, offer faster training, they may compromise on network performance.

Hybrid models, which amalgamate two or more forecasting methods, effectively surmount the limitations inherent in singular models. By harnessing the strengths of various methodologies, these combined models, such as SVM-LSTM [21], RF-LSTM [22], and SSA-SVR [23], significantly enhance prediction accuracy, representing a burgeoning trend in passenger flow forecasting. However, existing research predominantly focuses on optimizing model structures and training algorithm efficiency, often overlooking the impact of sample noise on model prediction performance. To mitigate noise interference and adeptly handle complex signals, experts have integrated filtering techniques, such as wavelet transform and empirical modal decomposition, into short-time subway passenger flow prediction models. Zhu et al. [24] developed a WT-ARMA combined prediction model utilizing wavelet transform to diminish noise in multimedia passenger flow data. Wu et al. [25] proposed a model combining variational modal decomposition (VMD) with GRU neural networks to enhance the accuracy of short-time metro passenger flow prediction by attenuating fluctuations in multimedia data. Similarly, Chen et al. [26] and Jo et al. [27] incorporated the season-trend decomposition procedure of time series by loess (STL) into LSTM and GRU neural network models, respectively, to improve short-term subway passenger flow prediction by counteracting the effects of irregular data fluctuations. Collectively, these models underscore the significance of employing filtering techniques to weaken the interference of noise in multimedia data samples in short-term traffic flow prediction.

3. Methodology

In order to improve the performance of the subway short-time passenger flow prediction model, a combined model based on two-stage decomposition (STL-EEMD) and GRU is constructed based on the characteristics of strong randomness and high nonstationarity of subway short-time passenger flow sequences in order to achieve higher prediction accuracy, thus providing theoretical support for subway operators to develop operation plans in advance. The proposed model consists of three parts: data preprocessing, data noise reduction, and passenger flow prediction. The model architecture is shown in Figure 1.

3.1. Data Preprocessing. Based on the swipe card data of the subway automatic ticketing system (AFC), the passenger flow in and out of the subway interchange station can be directly extracted. Since internal metro interchange does not require entry and exit stations, it is not possible to obtain interchange passenger flow information directly based on swipe card data. Therefore, the graph-based depth-first search algorithm is

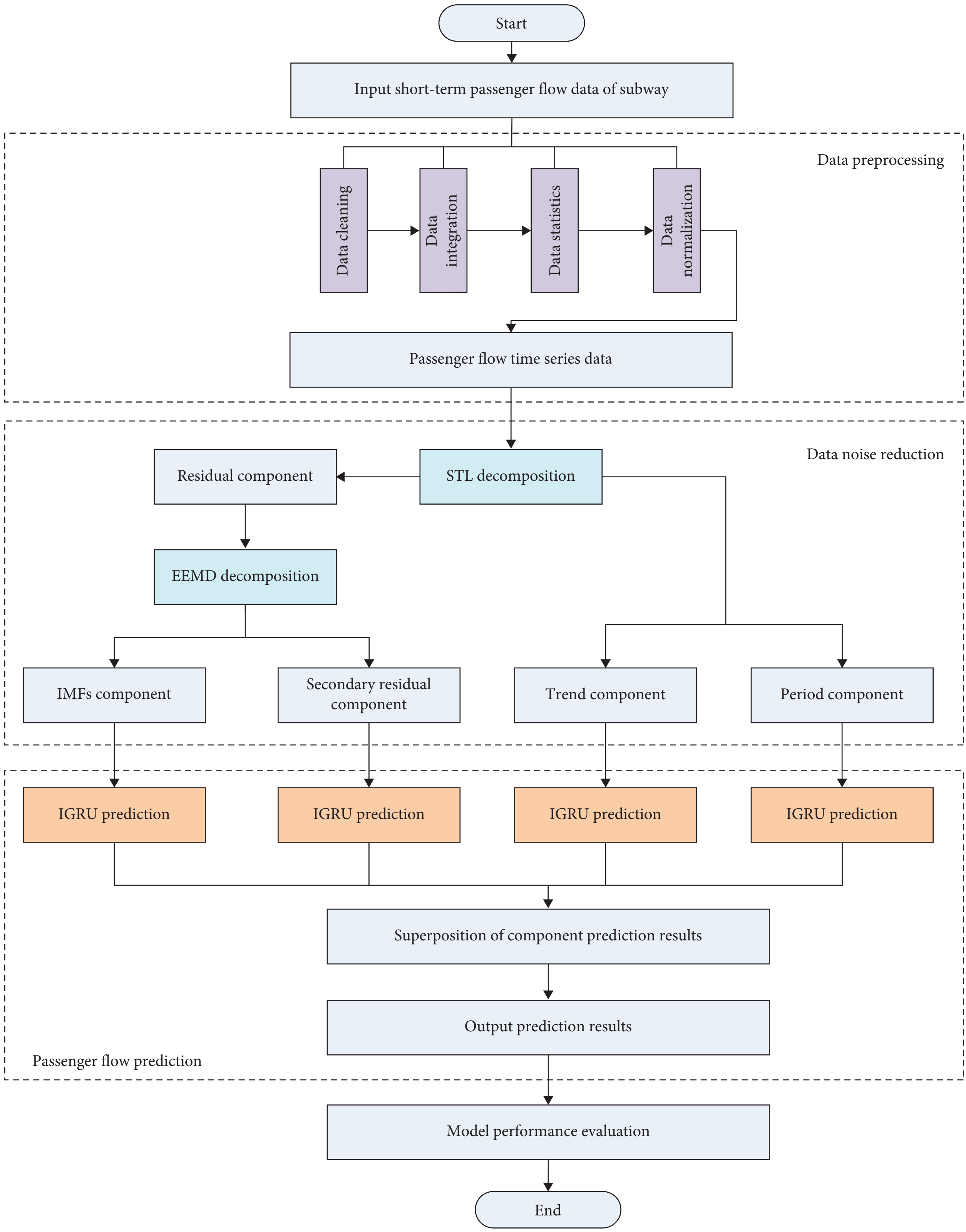

Figure 1: Architecture of the proposed metro short-time passenger flow prediction model.

Table 1: Main fields of the raw data of the metro AFC.

User card number	Inbound time	Exit time	Inbound route code	Inbound station code	Outbound route code	Outbound station code
84717573	2018-04-04 16:53:00	2018-04-04 18:47:03	1	23	90	29
15179225	2018-04-04 19:33:00	2018-04-04 20:55:45	5	62	6	15
14099543	2018-04-04 20:12:00	2018-04-04 20:51:07	3	37	11	36

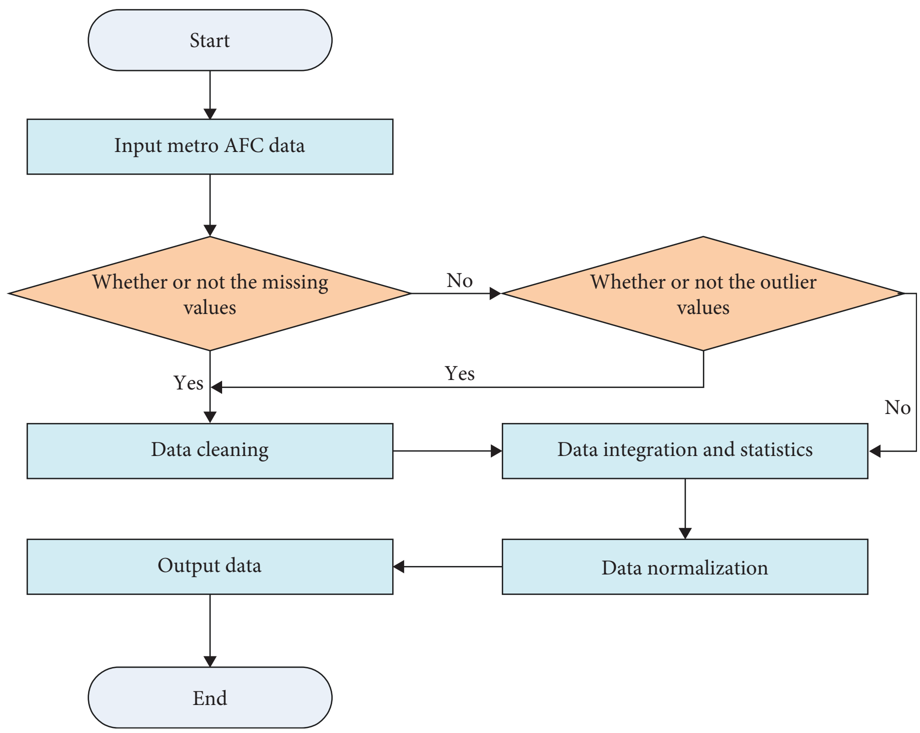

Figure 2: Flowchart of data preprocessing.

used to identify the travel path of subway passengers, which can more accurately extract the subway transfer passenger flow information and lay the foundation for the subway internal transfer passenger flow prediction.

The original data of metro AFC contains 43 fields, and the main fields are extracted, including user card number, entry and exit time, entry line and station code, exit line and station code, etc., as shown in Table 1.

In machine learning and data mining, the quality of multimedia data directly determines the prediction effect of the model. In real multimedia traffic data, there may be a large number of outliers, duplicate values, missing values, etc. This kind of data is very unfavorable to the training of neural network models. The purpose of data preprocessing is to obtain valid, standard, and continuous data for model training and data mining by processing invalid data accordingly. The preprocessed data can improve the learning speed of the model and make the model perform better in prediction results. Data preprocessing generally includes data cleaning, data integration, and data normalization processes. Figure 2 gives the data preprocessing flowchart.

3.1.1. Data Cleaning. The data-cleaning process is mainly cleans the outliers and missing values in multimedia passenger flow data. The time series model generally needs to ensure the integrity of the time series data, and if the missing values are directly removed, it will easily lead to the misalignment of data cycles. To ensure data integrity, the missing values in the original data need to be interpolated. During the data-cleaning process, we employed a time-series-based linear interpolation method to fill in missing values [28]. Specifically, we focused on interpolating missing values within timestamp information (such as entry and exit times) to ensure the continuity and consistency of the data. For discrete features (such as route/station codes), we did not perform interpolation but retained their original states.

In view of the problem of outliers in the original data, especially the data whose departure time is earlier than or equal to the arrival time, the card reading data outside the operation time of the subway line, and the data whose travel time is too long (more than 4 hr), the outliers are cleaned according to the subway AFC data cleaning rules.

3.1.2. Data Integration. Data integration is to make an organic concentration of data from different sources, formats, and characteristics of nature to facilitate subsequent statistical analysis of the overall data. Since the original swipe card data of the subway is a separate file for each day, it is necessary to use Python's pandas library to merge the original swipe card data and synthesize it into a data table to facilitate subsequent statistical analysis. In the short-time passenger flow prediction, it is necessary to make a division of the time granularity of the passenger flow, if the time granularity division is too small, it will affect the accuracy of the prediction. On the contrary, if the time granularity is too large, the prediction results will not reflect the change of short-time passenger flow. In this paper, we choose to use 15 min as the time granularity for passenger flow statistics.

3.1.3. Data Statistics. First, we need to split the time of the original data into three components: days, hours, and minutes, and then use 15 min as a unit for statistics. According to the schedule of subway operation for 1 day, the period of 06:00–22:30 is divided into 66 time periods, so the daily passenger flow data are divided into 66 time periods.

Second, the statistics of the passenger flow in and out of the subway stations are carried out. From the perspective of time, by dividing the time interval, we can count the passenger flow in and out of each station every 15 min. From the spatial perspective, we can count the passenger flow statistics of each station and line. The following data are integrated with the data to count the passenger flow in and out of the subway station every 15 min. In order to make the data closer to the real situation, the subway passenger flow during non-operating hours is filled with 0.

3.1.4. Data Normalization. The preprocessed inbound and outbound passenger flow data are still relatively large and require a lot of time to converge the model when it is put into model training, so the data need to be normalized. Data normalization is the process of scaling the valid data so that all the data fall within an interval required for model training. In order to speed up the convergence of the model, this paper uses the min–max normalization method to normalize the passenger flow in and out of subway stations, which is defined in Formula (1).

$$z = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{1}$$

where x denotes the passenger flow to and from all subway stations. The $\min(x)$ and $\max(x)$ denote the minimum and maximum values of the subway passenger flow, respectively.

3.2. Data Noise Reduction. The short-time passenger flow data of the metro has the characteristics of nonlinearity and strong randomness and contains a large amount of noise, which will reduce the accuracy of the prediction by direct passenger flow prediction. Therefore, this paper adopts the STL-EEMD method to reduce the interference of incoming passenger flow data noise. For the problem that the periodic terms obtained from the decomposition of the STL method are periodic terms with fixed amplitude, and for the multiple IMFs decomposed by EEMD, the trend terms and periodic terms cannot be accurately distinguished; this paper adopts a two-stage decomposition method for noise reduction. First, the STL method is used to decompose the time series to obtain the trend term, the periodic term, and the residual term. Second, the decomposed periodic and residual terms are decomposed again using the EEMD method.

3.2.1. STL Decomposition. STL is a time series decomposition method using locally weighted regression (LOESS) as a smoothing method, which has the advantages of simplicity of linear least squares regression and adaptability of nonlinear regression methods. The method decomposes the original time series into trend term, period term, and residual term, as shown in Formula (2).

$$Y_t = T_t + S_t + R_t \quad t = 1, \ldots m, \tag{2}$$

where Y_t denotes the original time series. T_t, S_t, and R_t denote the trend component, the periodic component, and the residual component at time t, respectively. In general, the trend term represents the trend of low-frequency variation, the period term represents the trend of high-frequency variation, and the residual component represents the irregular variation formed by random perturbations.

The STL method is a recursive process that requires three LOESS and a sliding average. The LOESS process is a locally weighted regression for different locations of points and different weights. This process assumes that it is based on the closer the distance, the stronger the correlation. It contains the window length, weight function, and order of the regression formula for selecting the local regression. Figure 3 shows the results of the STL method to decompose the raw weekday passenger flow data. In order to ensure that the period component accurately reflects the periodicity of the original time series, the number of periods of the STL time series decomposition needs to be chosen to be consistent with the number of periods of the original data; therefore, the period parameter is set to 66 in this study.

3.2.2. EEMD Decomposition. The EEMD method can effectively suppress the empirical modal decomposition aliasing phenomenon by adding white noise to the signal to be decomposed [29]. Therefore, the EEMD method is used to smooth the passenger flow time series data and decompose them into time series component data with different feature scales, spikes, and slower fluctuations, as shown in Figure 4.

The EEMD decomposition principle is as follows:

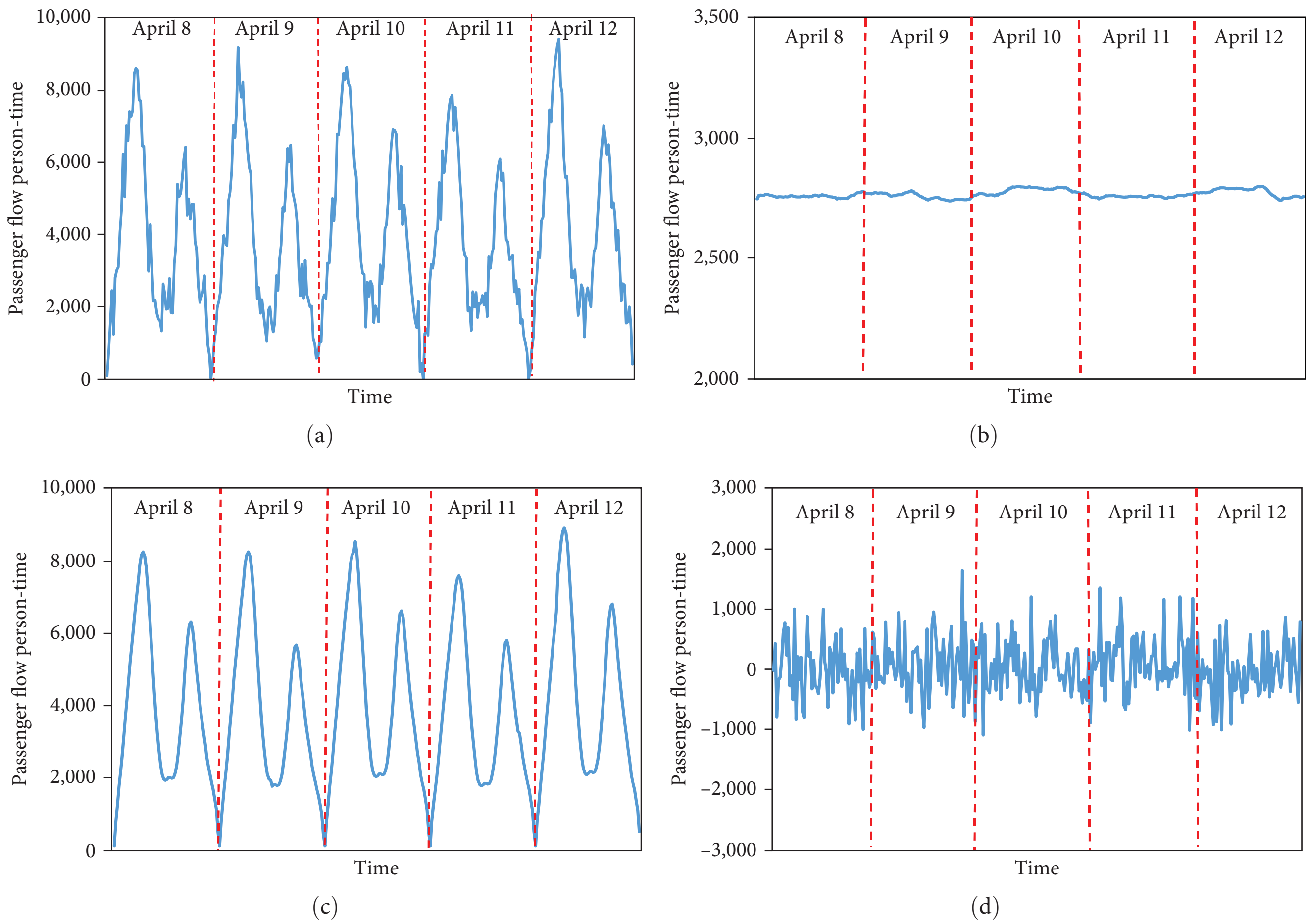

Figure 3: Results of STL method to decompose the raw data of passenger flow: (a) raw data; (b) trend component; (c) period data; (d) residual component.

(1) Add the normally distributed white noise signal to the original signal.

$$X'(t) = X(t) + \omega_j(t) \quad j = 1, 2, \ldots M, \tag{3}$$

where $X(t)$ is the original signal and $\omega_j(t)$ is the white noise signal. The $X'(t)$ is the generated new signal sequence. M is the number of tests.

(2) The new signal sequence is decomposed by EMD to obtain the IMF components.

$$X'(t) = \sum_{i=1}^{n} c_{i,j}(t) + r_{nj}(t), \tag{4}$$

where n is the number of IMF components obtained by EMD decomposition. $c_{i,j}(t)$ is the ith IMFs component in the jth experimental decomposition. $r_{nj}(t)$ is the residual component obtained from the decomposition.

(3) Repeat steps (1) and (2) above, adding white noise of different normal distribution each time.

(4) The average IMF component is obtained by averaging each IMF component.

$$c_i = \frac{1}{M} \sum_{j=1}^{M} c_{i,j}. \tag{5}$$

In the EEMD key parameters, the white noise standard deviation is set to 0.2, and the white noise count is set to 100.

3.3. Passenger Flow Forecasting Model. The trend, period, average IMF components, and quadratic residual residuals obtained from the above STL-EEMD decomposition are input into the IGRU neural network-based metro short-time passenger flow forecasting model for prediction.

3.3.1. GRU Network. LSTM solves the long-term dependency problem of (recurrent neural network) RNN, but it requires more parameters to be set, and the convergence speed is

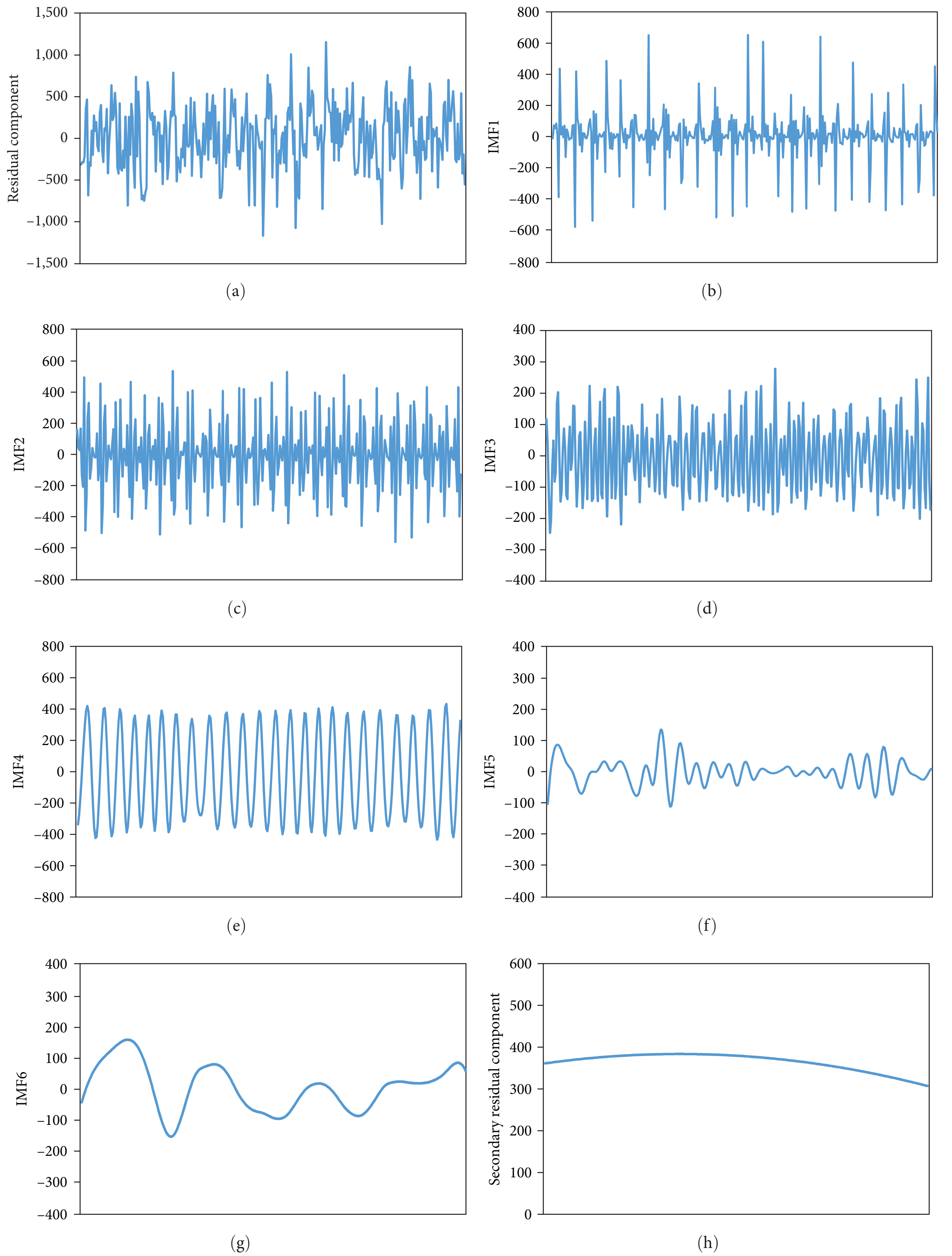

FIGURE 4: EEMD method decomposition results: (a) residual component; (b) IMF1; (c) IMF2; (d) IMF3; (e) IMF4; (f) IMF5; (g) IMF6; (h) secondary residual component.

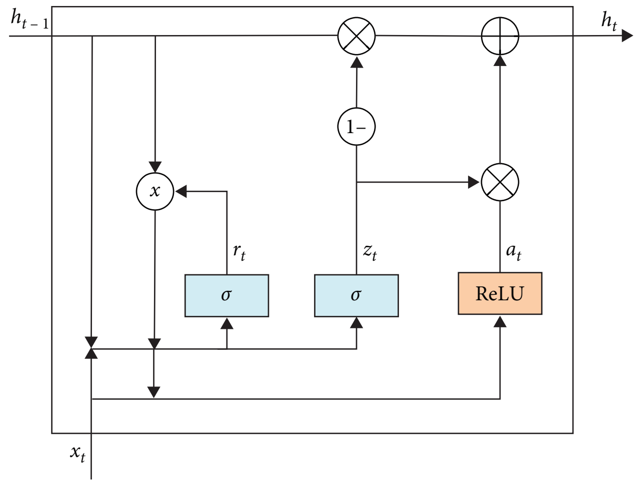

FIGURE 5: Schematic diagram of GRU unit structure.

slow, which reduces the training efficiency. GRU neural network is an improved version of LSTM, and the update gate replaces the input gate and forgetting gate in LSTM. The structure of the hidden layer of the GRU network is shown in Figure 5.

The update gate is used to determine the amount of information passed from the previous hidden layer to the current hidden layer, and the reset gate determines the amount of information about the forgotten state. The GRU cell structure works as expressed in the following formulas:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z), \tag{6}$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \tag{7}$$

$$a_t = \mathrm{Tan}h\,(W_a U_a (h_{t-1} r_t) + b_a), \tag{8}$$

$$h_t = (1 - z_t) h_{t-1} + z_t a_t, \tag{9}$$

where x_t denotes the input value at time t of the current layer, and h_{t-1} denotes the state output value at time $t-1$ of the current layer. z_t and r_t denote the update gate and reset gate at time t, respectively. Sigmoid functions are used for the activation functions of update and reset gates σ. a_t denotes the candidate hidden state at time t. h_t denotes the state vector at time t. Tan$h$ is the hyperbolic tangent activation function for the candidate hidden state. w_z, w_r, w_a, u_z, u_r, and u_a denote the model weight parameters. b_z, b_r, and b_a denote the bias vectors.

3.3.2. IGRU Network. To address the gradient disappearance and network degradation of the original GRU network, a residual-based gated cyclic unit is designed in this paper, as shown in Figure 6. Compared with the GRU unit, the residual gated loop unit is improved in the following three aspects:

(1) Nonsaturated Activation Function. The core formula in the algorithm of GRU is the candidate hidden state Formula (8). The output value of Formula (8) and the output value of the previous hidden state together determine the final output of the GRU hidden state. In this paper, the

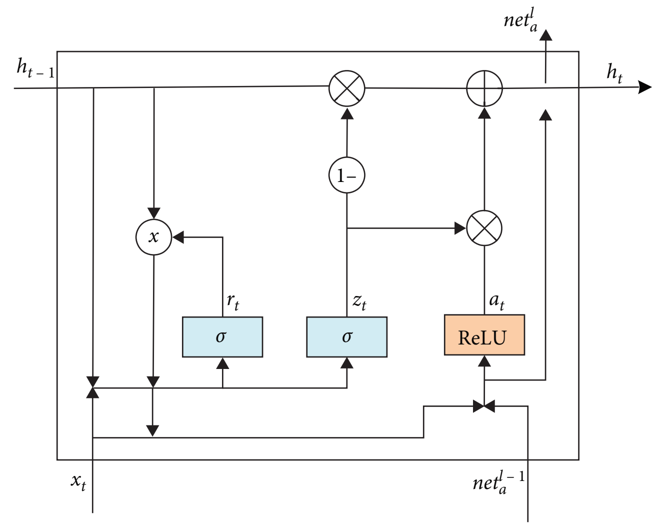

FIGURE 6: Schematic diagram of the structure of the residual gating cycle unit.

activation function of the candidate hidden state of GRU is replaced by the linear rectification function ReLU, which, the advantage of the improved network, can well avoid the gradient disappearance caused by the saturation function. It can cope with the deeper network training. The ReLU function is defined as follows:

$$\mathrm{ReLU}(f(x)) = \begin{cases} f(x), & f(x) \geq 0 \\ 0, & f(x) < 0 \end{cases}. \tag{10}$$

The ReLU activation function ensures a more direct information transfer. Compared with saturated activation functions, ReLU does not have the gradient disappearance problem caused by saturated activation functions, and it can better match the residual information transfer. Therefore, Formula (8) can be changed to the following:

$$a_t = \mathrm{ReLU}(W_a x_t + U_a (h_{t-1} r_t) + b_a). \tag{11}$$

In traditional RNN neural networks, the use of unsaturated activation functions without boundaries usually generates the gradient explosion problem. ReLU, a representative of unsaturated activation functions, also suffers from the gradient explosion problem. The gradient explosion problem can be effectively mitigated by combining unsaturated activation functions with batch normalization techniques [30].

(2) *Residual Connection*. In this paper, the GRU is improved by referring to the residual network in the convolutional neural network to solve the problem of gradient disappearance and network degradation in the GRU. Specifically, we put the residual connection in Formula (11). For the introduced residual information, we use the candidate hidden state values that are not yet activated in the previous layer, because the unactivated values have more original information than the activated ones. In addition, unlike the residual network in the convolutional neural network, the

improved scheme designed in this paper introduces residual connections into each layer of the GRU. The improved hidden state formula is as follows:

$$a_t^l = \mathrm{ReLU}\left(\mathrm{net}_a^l\right), \tag{12}$$

$$net_a^l = \left(W_a^l x_t^l + U_a^l\left(h_{t-1}^l r_t^l\right) + b_a^l\right) + V^l net_a^{l-1}, \tag{13}$$

where a_t^l denotes the output of the candidate hidden states of layer l at moment t. net_a^l is the candidate hidden state of layer l that has not yet been activated, and h_{t-1}^l denotes the state vector of layer l at time $t-1$, and V^l is the dimensional matching matrix of the lth layer. When the dimensionality of the upper and lower layers of the network is the same, the dimensionality matching matrix is not needed.

(3) *Batch Standardization*. Batch normalization addresses the internal covariance bias of the data by normalizing the mean and variance of the preactivation for each layer of each training minibatch, and also accelerates training engineering and improves system performance. In addition, the use of batch normalization can alleviate the gradient explosion problem caused by unsaturated activation functions. In this paper, by changing the activation function of GRU and adding residual connections, and then using the advantages of batch normalization, we can eliminate the gradient disappearance and network degradation in traditional GRU.

The cell structure of the gated cyclic cell at level l, combining the residuals after batch normalization, is given by the following:

$$z_t^l = \sigma\left(\mathrm{BN}\left(W_z^l x_t^l\right) + U_z^l h_{t-1}^l\right), \tag{14}$$

$$r_t^l = \sigma\left(\mathrm{BN}\left(W_r^l x_t^l\right) + U_r^l h_{t-1}^l\right), \tag{15}$$

$$a_t^l = \mathrm{ReLU}\left(net_a^l\right), \tag{16}$$

$$net_a^l = \mathrm{BN}\left(W_a^l x_t^l\right) + U_a^l\left(h_{t-1}^l r_t^l\right) + V^l net_a^{l-1}, \tag{17}$$

$$h_t^l = \left(1 - z_r^l\right) h_{t-1}^l + z_r^l a_t^l\big), \tag{18}$$

where BN denotes the batch normalization used. Since the nature of batch normalization is to eliminate bias, the bias vectors in Formulas (14), (15), and (17) are neglected.

3.3.3. Loss Function for IGRU. The loss function for IGRU plays a crucial role in guiding the training process of the model and optimizing its performance in short-time passenger flow forecasting tasks. Unlike standard RNN that often employ common loss functions such as mean squared error (MSE) or cross-entropy loss, the formulation of the loss function for IGRU involves specific considerations tailored to its architecture and objectives.

Mathematically, the loss function for IGRU is formulated as a function of the model's parameters (weights and biases) and the discrepancy between the predicted and actual passenger flow values. By minimizing this loss function using optimization algorithms such as stochastic gradient descent (SGD) or Adam, the model iteratively adjusts its parameters to improve prediction accuracy and overall performance.

The loss function L_{IGRU} for IGRU is defined as MSE between the predicted passenger flow values $\widehat{y}_t$ and the actual passenger flow values y_t over a given time horizon T:

$$L_{\mathrm{IGRU}} = \frac{1}{T}\sum_{t=1}^{T}\left(\widehat{y}_t - y_t\right)^2, \tag{19}$$

where $\widehat{y}_t$ represents the predicted passenger flow value at time step t; y_t denotes the actual passenger flow value at time step t; T denotes the total number of time steps in the prediction horizon.

By minimizing the loss function L_{IGRU} through SGD or Adam, the parameters of the IGRU model are adjusted iteratively to improve the accuracy of passenger flow predictions.

3.4. The Process of Constructing a Passenger Flow Prediction Model. Based on the above analysis, the metro short-time passenger flow prediction model is constructed as follows.

3.4.1. Data Preprocessing. We obtain valid, standardized, and continuous model training data through data cleaning, integration, statistics, and normalization for a large number of outliers, duplicate values, and missing values in passenger flow data.

3.4.2. Data Noise Reduction. The STL decomposition algorithm is used to decompose the preprocessed data into trend terms, periodic terms, and irregular fluctuation terms. Based on this, the EEMD decomposition algorithm decomposes the irregular fluctuation terms again to obtain multiple IMF components and RES residual residuals.

3.4.3. Passenger Flow Model Forecast. The STL and EEMD decomposed data are divided into training set, test set, and normalized. The training set data is used to train the model to obtain the best performance of the subway short-time passenger flow prediction model. The trained passenger flow prediction model is used to predict the test set data, and the prediction results are reverse normalized to obtain the prediction results of each decomposition component.

3.4.4. Component Superposition. The prediction results of each component are superimposed and summed to obtain the final short-time passenger flow prediction results.

3.4.5. Model Effectiveness Evaluation. Appropriate evaluation parameters are selected to assess the prediction model effects.

4. Result Analysis and Discussion

4.1. Experimental Data. In view of the availability of data, this paper selects Shanghai Metro automatic card swipe data for one consecutive month from April 1 to 30, 2015 and preprocesses the missing and abnormal values. The metro automatic ticketing data contains travel information such as the card number used by passengers, the name of entering and leaving the station, the time of entering and leaving the station, and the fare, which provides data support for the

Table 2: Correlation analysis of workday and nonworkday data.

Correlation sequence	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
Monday	1	0.9941	0.9936	0.9923	0.9887	0.5418	0.4932
Tuesday	0.9941	1	0.9936	0.9949	0.9911	0.5537	0.5279
Wednesday	0.9936	0.9936	1	0.9938	0.9907	0.5665	0.5193
Thursday	0.9923	0.9949	0.9938	1	0.9922	0.5631	0.5209
Friday	0.9887	0.9911	0.9907	0.9922	1	0.6208	0.5837
Saturday	0.5418	0.5537	0.5665	0.5631	0.6208	1	0.9235
Sunday	0.4932	0.5279	0.5193	0.5209	0.5837	0.9235	1

study of passenger travel patterns. As of April 2015, Shanghai Metro has operated 14 metro lines with 313 stations (interchange stations are not counted repeatedly). People's Square Station is a three-line interchange station of metro lines 1, 2, and 8, with large passenger flow and complex travel characteristics; therefore, People's Square Station is chosen as the target station. In this paper, the prediction step is taken as 15 min, and the prediction time range is 6:00–22:30, which means there are 66 prediction values per day.

There is some similarity in the change of passenger flow on weekdays from Monday to Friday, and also on nonworking days on Saturday. The study was based on passenger flow data for 1 week (April 6–12, 2015), and a correlation analysis was conducted. The correlation analysis of the weekday and nonworkday data is shown in Table 2.

As can be seen from Table 2, the correlation between weekday data and the correlation between nonworkday data are significant and are both highly correlated. There are large differences between nonworkday and workday data, which are moderate to low correlations. Therefore, the passenger flow forecast is studied from two perspectives, weekday and nonworkday. Therefore, the data from 20 out of 21 weekdays during April 1–30, 2015, were used as the training set, and data from 1 weekday were used as the test set. The data from 8 out of 9 nonworking days were used as the training set, and data from 1 nonworking day were used as the test set.

4.2. Model Parameters and Evaluation Indexes. In this paper, the IGRU network is set up with 2 hidden layers, 1 input layer and 1 output layer, with 10 neurons in the input layer, 1 neuron in the output layer, and 50 neurons in the hidden layer. The number of iterations is 100, and the batch size is 8. To avoid overfitting of the model due to the high specialization of neuron weights, a dropout layer is added, and the random deactivation probability is set to 0.1. The mean absolute error MAE is chosen as the target loss function. The Adam optimizer is used, which customizes the initial learning rate to 0.001 and automatically updates the learning rate of each parameter every round by an adaptive method.

In this paper, mean absolute percentage error (MAPE) and root mean square error (RMSE) are used as the evaluation indexes of the forecasting model. MAPE reflects the relative deviation of the observed value from the true value and can directly measure the goodness of the forecasting result, which is defined as follows:

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \widehat{y}_i|}{y_i} \times 100\%, \tag{20}$$

where y_i and $\widehat{y}_i$ are the ith actual observation and the predicted value, respectively. n is the total number of predictions. MAPE, which is often used to evaluate the merit of prediction models, does not directly reflect the difference between the observed and true values.

The RMSE directly reflects the absolute difference between the observed and true values and is very sensitive to the reflection of very large or very small errors, and is a useful complement to MAPE when comparing model prediction accuracy, which is defined as follows:

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y}_i)^2}. \tag{21}$$

4.3. Comparison Experiments. To verify the effectiveness of the proposed model, the model was tested using experimental data, and the results were compared with advanced models such as ARIMA [13], BPNN [13], GRU [20], VMD-GRU [25], STL-LSTM [26], and STL-GRU [27]. The prediction results of several forecasting models for weekdays and nonworking days are given in Figures 7 and 8, where the horizontal coordinates represent the time of 1 day divided by 15 min.

In order to quantitatively evaluate the performance of the models, two evaluation indexes, MAPE and RMSE, are used to compare and analyze several forecasting models. The values of MAPE and RMSE are the average values of short-time passenger flow forecasting models after 10 independent runs, and the test results of several forecasting models are shown in Figures 9 and 10.

As can be seen from Figures 9 and 10, the error values of the proposed STL-EEMD-IGRU model are smaller than the other six forecasting models. Compared with the single model ARIMA, BPNN, and GRU models, the advantage of the combined model is very obvious, and the errors are smaller than those of the single model on both weekdays and nonweekdays. For the combined model, the STL-EEMD-IGRU model has the smallest error value and the highest prediction accuracy because the model proposed in this paper has appropriate improvements in both data noise reduction and passenger flow prediction models. In particular, compared with the STL-GRU model, the MAPE is

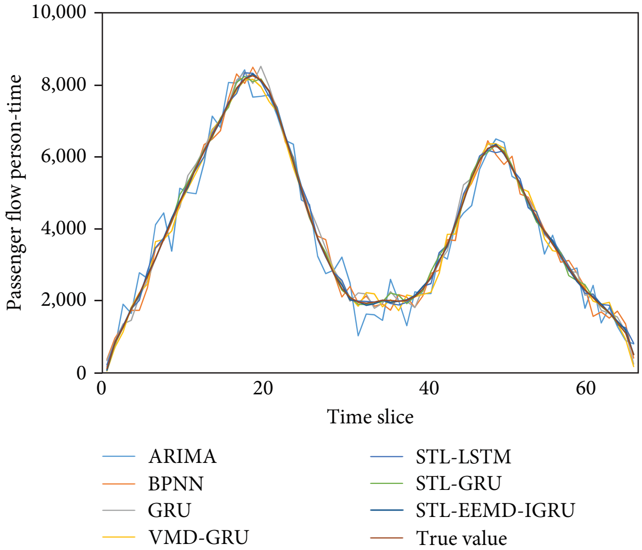

Figure 7: Comparison of forecast results for working days.

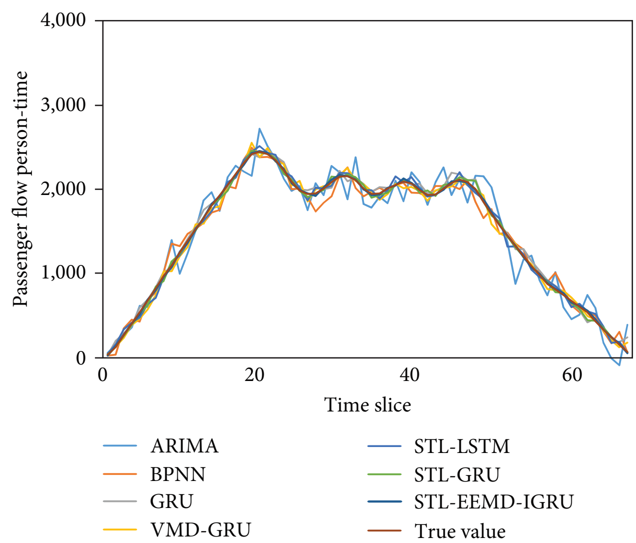

Figure 8: Comparison of forecast results for nonworking days.

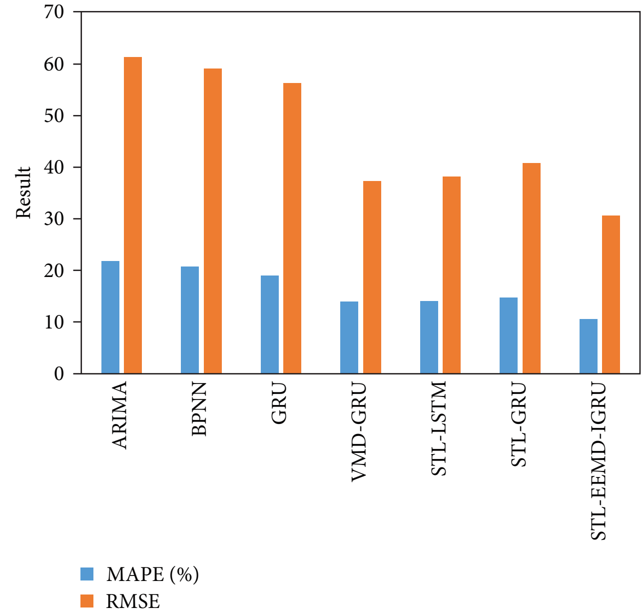

Figure 9: Comparison of prediction errors for working days.

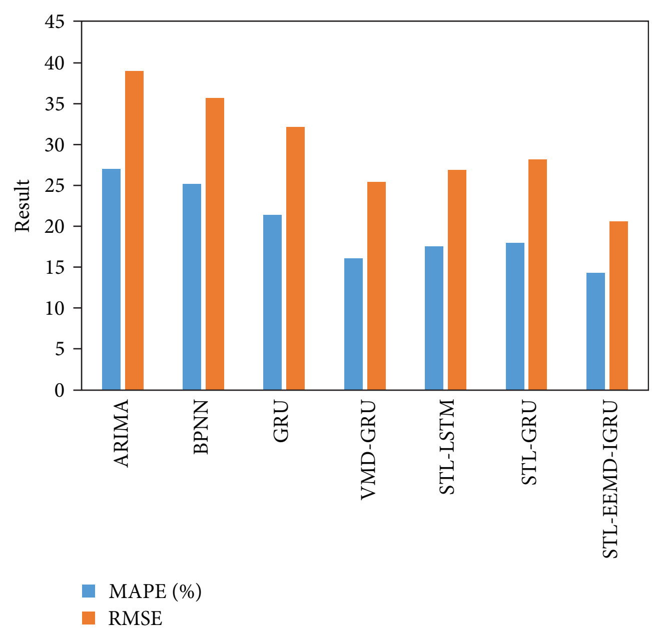

Figure 10: Comparison of forecast errors for nonworking days.

reduced by 28.34%, and the RMSE is reduced by 25.01% in the weekday passenger flow prediction. The EMD-PSO-LSTM model reduces AMAPE and ARMSE by 20.36% and 26.84%, respectively, compared with the STL-GRU model for nonworking day passenger flows.

4.4. Ablation Experiments. In order to investigate the proposed noise suppression technique and the improved GRU algorithm in the proposed model, ablation experiments were performed on the model using the workday data from the experimental data and compared using MAPE and RMSE metrics. The comparison graph of the ablation experiment is shown in Figure 11.

It can be seen from Figure 11 that the contribution of noise suppression technology is greater than that of the IGRU network to the proposed model. The reason for this is that the strong randomness and high nonstationarity of the metro short-time passenger flow data noise are particularly disturbing to the prediction model, and only by decomposing the original time series into a set of relatively simple submodal smooth fluctuations, higher prediction accuracy can be obtained. The forecasting capability and robustness of the model can be improved. In addition, the residual-based IGRU network mainly targets the gradient disappearance and network degradation of the original GRU network. The more layers of the GRU network, the better the effect, and only two hidden layers are used in this paper, so the effect of the IGRU network is not outstanding.

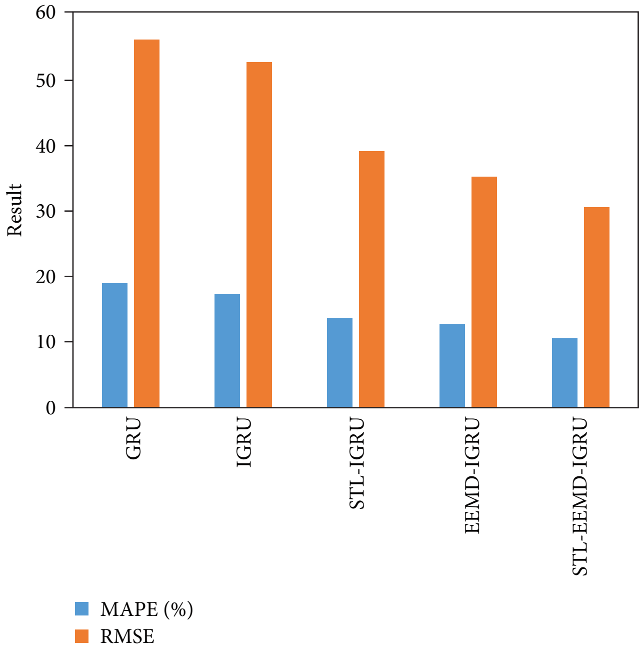

FIGURE 11: Comparison chart of ablation experiment.

5. Conclusion

This research introduces an advanced metro short-time passenger flow prediction methodology employing a combined STL-EEMD-IGRU model. This approach aims to enhance the predictive accuracy of short-time passenger flow in urban rail transit systems. In the realm of machine learning and data mining, the integrity and quality of multimedia data are pivotal for the effectiveness of predictive models. Given the potential presence of outliers, duplicate values, and missing data in real-world multimedia passenger flow datasets, an initial phase of data preprocessing is essential in the application of the STL-EEMD-IGRU combined model.

Subsequently, the STL-EEMD two-stage decomposition technique is utilized to attenuate noise interference within the short-time passenger flow time series, thereby diminishing the impact of sample noise on the passenger flow prediction model. Following this noise reduction, the data are further processed through a GRU neural network.

A critical aspect of this research is addressing the challenges of gradient vanishing and network degradation commonly associated with conventional GRU networks. To this end, the study proposes a novel design of a residual-based gating cycle unit aimed at bolstering the predictive performance of the network model.

Empirical evaluations of this combined forecasting model demonstrate its effectiveness, particularly in enhancing the accuracy of the subway short-term passenger flow prediction. The outcomes of this study offer valuable data-driven insights for subway operation management departments, facilitating improved passenger flow management at stations and the development of more efficient daily traffic plans.

Data Availability

The labeled data set used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the construct program of applied characteristic discipline in Hunan Province and the Project of Hunan Provincial Natural Science Foundation of China (grant no. 2023JJ50421).

References

[1] L. N. N. Do, N. Taherifar, and H. L. Vu, “Survey of neural network-based models for short-term traffic state prediction,” *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 1, Article ID e1285, 2019.

[2] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, “A noise-immune Kalman filter for short-term traffic flow forecasting,” *Physica A: Statistical Mechanics and Its Applications*, vol. 536, pp. 122601–122609, 2019.

[3] S. Cheng, F. Lu, and P. Peng, “Short-term traffic forecasting by mining the non-stationarity of spatiotemporal patterns,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6365–6383, 2021.

[4] J. Tang, L. Yang, and W. Yang, “Urban rail transit short-term passenger flow origin-destination forecast based on LSTM and GCN,” *Journal of Chongqing University*, vol. 45, no. 11, pp. 91–99, 2022.

[5] S. Lu, Q. Zhang, G. Chen, and D. Seng, “A combined method for short-term traffic flow prediction based on recurrent neural network,” *Alexandria Engineering Journal*, vol. 60, no. 1, pp. 87–94, 2021.

[6] L. Cai, Q. Chen, W. Cai, X. Xu, T. Zhou, and J. Qin, “SVRGSA: a hybrid learning based model for short-term traffic flow forecasting,” *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1348–1355, 2019.

[7] Q. Zhang, H. Qian, Y. Chen, and D. Lei, “A short-term traffic forecasting model based on echo state network optimized by improved fruit fly optimization algorithm,” *Neurocomputing*, vol. 416, pp. 117–124, 2020.

[8] Y. Gu, W. Lu, X. Xu, L. Qin, Z. Shao, and H. Zhang, “An improved Bayesian combination model for short-term traffic prediction with deep learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1332–1342, 2019.

[9] S. O. Mousavizadeh Kashi and M. Akbarzadeh, “A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks,” *Journal of Intelligent Transportation Systems*, vol. 23, no. 1, pp. 60–71, 2019.

[10] L. Zeng, Z. Li, J. Yang, and X. Xu, “CEEMDAN-IPSO-LSTM: a novel model for short-term passenger flow prediction in urban rail transit systems,” *International Journal of Environmental Research and Public Health*, vol. 19, no. 24, pp. 16433–16451, 2022.

[11] M. Zhao, W. Zhang, K. Wang, and H. Li, "Short-term passenger flow prediction of urban rail transit based on EMD-PSO-LSTM combined model," *Railway Transport and Economy*, vol. 44, no. 7, pp. 110–118, 2022.

[12] S. Zhang, Y. Song, D. Jiang, T. Zhou, and J. Qin, "Noise-identified Kalman filter for short-term traffic flow forecasting," in *2019 15th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*, pp. 462–466, IEEE, Shenzhen, China, 2019.

[13] Y. Ye, L. Chen, and F. Xue, "Passenger flow prediction in bus transportation system using ARIMA models with big data," in *2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC)*, pp. 436–443, IEEE, Guilin, China, 2019.

[14] Z. Song, Y. Guo, Y. Wu, J. Ma, and J. Tang, "Short-term traffic speed prediction under different data collection time intervals using a SARIMA-SDGM hybrid prediction model," *PLOS ONE*, vol. 14, no. 6, Article ID e0218626, 2019.

[15] J. Li, F. Guo, A. Sivakumar, Y. Dong, and R. Krishnan, "Transferability improvement in short-term traffic prediction using stacked LSTM network," *Transportation Research Part C: Emerging Technologies*, vol. 124, pp. 102977–103003, 2021.

[16] S. Wang and X. Yang, "Short-term passenger flow forecasting of public transport based on EEMD-GWO-LSSVM," *Computer Engineering and Applications*, vol. 55, no. 20, pp. 216–221 239, 2019.

[17] J. Zhao, J. Jiang, W. Liu, Y. Xu, and M. Li, "The short-term passenger flow forecast of Xi'an rail transit based on BP neural network," *International Core Journal of Engineering*, vol. 8, no. 6, pp. 192–200, 2022.

[18] Z. Cheng, J. Lu, H. Zhou, Y. Zhang, and L. Zhang, "Short-term traffic flow prediction: an integrated method of econometrics and hybrid deep learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5231–5244, 2022.

[19] J. Ma, X. Zeng, X. Xue, and R. Deng, "Metro emergency passenger flow prediction on transfer learning and LSTM model," *Applied Sciences*, vol. 12, no. 3, pp. 1644–1657, 2022.

[20] G. Dai, C. Ma, and X. Xu, "Short-term traffic flow prediction method for urban road sections based on space-time analysis and GRU," *IEEE Access*, vol. 7, pp. 143025–143035, 2019.

[21] J. Guo, Z. Xie, Y. Qin, L. Jia, and Y. Wang, "Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM," *IEEE Access*, vol. 7, pp. 42946–42955, 2019.

[22] S. Lin and H. Tian, "Short-term metro passenger flow prediction based on random forest and LSTM," in *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, vol. 1, pp. 2520–2526, IEEE, Chongqing, China, 2020.

[23] C. Shuai, Y. Xie, J. Shan, and Y. Ou, "Prediction of short-term inbound passenger flow of urban rail transit based on the singular spectrum analysis and support vector regression model," *Urban Rapid Rail Transit*, vol. 35, no. 5, pp. 76–83, 2022.

[24] C. Zhu, X. Sun, P. Li, J. Zhang, and Y. Li, "Short-term urban rail transit passenger flow prediction based on incorporating station classification and data noise reduction," *Journal of Railway Science and Engineering*, vol. 19, no. 8, pp. 2182–2192, 2022.

[25] J. Wu, Y. He, N. Zhang, and H. Wu, "Forecast of short-term passenger flow of urban rail transit based on VMD-GRU," *Urban Rapid Rail Transit*, vol. 35, no. 1, pp. 79–86, 2022.

[26] D. Chen, J. Zhang, and S. Jiang, "Forecasting the short-term metro ridership with seasonal and trend decomposition using loess and LSTM neural networks," *IEEE Access*, vol. 8, pp. 91181–91187, 2020.

[27] H. J. Jo, W. J. Kim, H. K. Goh, and C.-H. Jun, "An improved time-series forecasting model using time series decomposition and GRU architecture," in *Neural Information Processing. ICONIP 2021*, vol. 1517 of *Communications in Computer and Information Science*, pp. 587–596, Springer, Cham, 2021.

[28] C. Oh, S. Han, and J. Jeong, "Time-series data augmentation based on interpolation," *Procedia Computer Science*, vol. 175, pp. 64–71, 2020.

[29] Y. Cao, X. Hou, and N. Chen, "Short-term forecast of OD passenger flow based on ensemble empirical mode decomposition," *Sustainability*, vol. 14, no. 14, pp. 8562–8575, 2022.

[30] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Light gated recurrent units for speech recognition," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 2, pp. 92–102, 2018.