



Article

CEEMDAN-IPSO-LSTM: A Novel Model for Short-Term Passenger Flow Prediction in Urban Rail Transit Systems

Lu Zeng^{1,2,3,*} , Zinuo Li¹, Jie Yang^{1,3} and Xinyue Xu²

¹ School of Electrical Engineering, Jiangxi University of Science and Technology, Ganzhou 341000, China

² State Key Lab of Rail Traffic Control and Safety, Beijing Jiaotong University, Beijing 100044, China

³ Ganjiang Innovation Academy, Chinese Academy of Sciences, Ganzhou 341000, China

* Correspondence: 13114227@bjtu.edu.cn

Abstract: Urban rail transit (URT) is a key mode of public transport, which serves for greatest user demand. Short-term passenger flow prediction aims to improve management validity and avoid extravagance of public transport resources. In order to anticipate passenger flow for URT, managing nonlinearity, correlation, and periodicity of data series in a single model is difficult. This paper offers a short-term passenger flow prediction combination model based on complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) and long-short term memory neural network (LSTM) in order to more accurately anticipate the short-period passenger flow of URT. In the meantime, the hyperparameters of LSTM were calculated using the improved particle swarm optimization (IPSO). First, CEEMDAN-IPSO-LSTM model performed the CEEMDAN decomposition of passenger flow data and obtained uncoupled intrinsic mode functions and a residual sequence after removing noisy data. Second, we built a CEEMDAN-IPSO-LSTM passenger flow prediction model for each decomposed component and extracted prediction values. Third, the experimental results showed that compared with the single LSTM model, CEEMDAN-IPSO-LSTM model reduced by 40 persons/35 persons, 44 persons/35 persons, 37 persons/31 persons, and 46.89%/35.1% in SD, RMSE, MAE, and MAPE, and increase by 2.32%/3.63% and 2.19%/1.67% in R and R², respectively. This model can reduce the risks of public health security due to excessive crowding of passengers (especially in the period of COVID-19), as well as reduce the negative impact on the environment through the optimization of traffic flows, and develop low-carbon transportation.

Keywords: urban rail transit; short-term passenger flow prediction; complete ensemble empirical mode decomposition with adaptive noise; long-short term memory neural network; improved particle swarm optimization; combination model; CEEMDAN-IPSO-LSTM



Citation: Zeng, L.; Li, Z.; Yang, J.; Xu, X. CEEMDAN-IPSO-LSTM: A Novel Model for Short-Term Passenger Flow Prediction in Urban Rail Transit Systems. *Int. J. Environ. Res. Public Health* **2022**, *19*, 16433. <https://doi.org/10.3390/ijerph192416433>

Academic Editor: Paul B. Tchounwou

Received: 9 November 2022

Accepted: 2 December 2022

Published: 7 December 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The influence of human activities on the global climate, characterized by global warming, has had serious negative impacts on public health. Energy conservation and carbon reduction have become serious environmental development issues to address. At the 75th United Nations General Assembly on 22 September 2020, China announced it would reach a peak in CO₂ emissions by 2030 and achieve carbon neutrality before 2060 (hereinafter referred to as double carbon goals) [1].

With the continuous improvement of China's urbanization level and the diversification of urban transport logistics and travel demand, the transport sector has become the main body of China's energy consumption and carbon emissions growth [2]. A key strategy for lowering urban carbon emissions is the expansion of public transportation [3,4]. Urban rail transit (hereinafter referred to as URT) is a large-capacity public transport infrastructure and the backbone of low-carbon transportation in cities. The URT in China has been rapidly increasing, and its energy consumption and carbon emission reduction pressure remains high. As of 30 September 2022, 52 mainland Chinese cities have put into operation

9788.64 km of URT lines, including 7655.32 km of subway, accounting for 78.21% [5]. Passenger flow volume is rapidly growing along with URT's quick expansion, which is producing severe congestion in URT systems. Accurately predicting the short-term flow volume and subsequently carrying out the necessary management procedures are two ways by which to relieve traffic congestion [6,7]. Travelers can effectively change their preferred method of transportation, route, or travel dates in advance by properly forecasting the influx and outflow of each station in a URT, which reduces travel time and costs [8,9]. Utilizing the prediction data, operators can identify crowded stations. The relevant passenger control measures can be put in place at stations that are severely congested to prevent congestion. Moreover, the timetable can be timely optimized so as to transport more passengers during peak hours according to predictions results.

At present, the research on short-time passenger flow prediction of URT at home and abroad is mainly conducted through three categories: statistical methods, traditional machine learning methods, and deep learning methods. Statistical methods are more sensitive to the linear relationship between variables, but they cannot capture the nonlinear relationship in the data. Such methods mainly include Kalman Filter model [10,11], ARMA model [12], and ARIMA model [13–15]. Traditional machine learning methods can better capture the nonlinear features in time series, and the accuracy for rail transit passenger flow prediction is higher. Such methods mainly include Support Vector Machine [16,17] and neural network [18–20]. However, the prediction model using traditional machine learning methods is prone to over-learning or under-learning problems when dealing with massive passenger flow data, which affects the accuracy of prediction models [21]. With the advancement of related theories and technologies, researchers have begun to use deep learning models to predict URT passenger flow [22]. Due to the strong applicability of the LSTM model in processing time series data, it has been widely used in passenger flow forecasting research [23–25].

The achievement of a single model's good prediction performance in real-world case studies is undoubtedly difficult. As a result, more academics have increasingly concentrated on combination forecasting models. Gong et al. [26] set up a passenger flow forecasting framework combining the seasonal ARIMA-based method and Kalman filter-based method. The framework was applied to the real bus line for passenger flow prediction. Qin et al. [27] coupled a seasonal-trend decomposition approach with an adaptive boosting framework to anticipate the monthly passenger flow on China Railway. A prediction model for irregular passenger flow based on the combination of support vector regression and LSTM was presented by Guo et al. [28]. A three-stage passenger flow forecasting model was developed by Liu and Chen [29] using a deep neural network and stacked automated encoder. The performance of the prediction was shown to be significantly impacted by the choice and combination of important features.

Although the accuracy of the aforementioned prediction methods has somewhat increased, neither the interference of passenger flow data noise nor the manual trial-and-error method of determining the hyperparameters of the neural network based solely on empirical values has been considered. In order to address these issues, this paper combines the CEEMDAN algorithm for reducing data noise interference with the IPSO algorithm for hyperparameters optimization of LSTM neural networks to create a new short-term passenger flow prediction model of URT based on CEEMDAN-IPSO-LSTM. The model's predictive performance is then thoroughly assessed using the benchmark function, prediction error, and Taylor diagram. In a word, short-term passenger flow accurate prediction of URT can improve the efficiency of transport infrastructure and means of transport. At the same time, it can further put forward optimization suggestions for URT operation management during the post-epidemic period, and provide a reference for the early realization of the dual carbon goals.

2. Methods

2.1. CEEMDAN Algorithm

The complete ensemble empirical mode decomposition with adaptive noise (CEEMDAN) algorithm is a time-frequency domain analysis method that excels at nonlinear and non-stationary data due to its excellent adaptivity and convergence [30]. Through the addition of adaptive noise, the modal effects are further diminished. This algorithm can decompose complex time series data into intrinsic modal functions (IMFs) and a residual (Res), so as to effectively solve problems such as boundary effects and low computational efficiency that EMD [31], EEMD [32], and CEEMD [33] are prone to.

The following are the specific steps of the CEEMDAN algorithm.

$x(t)$ is the original passenger flow time series; $\overline{IMF}_k(t)$ is the k th IMF obtained by CEEMDAN decomposition; $EMD_j(*)$ represents the j th IMF obtained by EMD decomposition; $\beta_k (k = 2, 3, \dots, K)$ is a scalar coefficient that is used to adjust the signal-to-noise ratio at each stage, determining the standard deviation of the Gaussian white noise in the process; $\omega_i(t) (i = 1, 2, \dots, n)$ is the Gaussian white noise that adheres to the standard normal distribution.

Step 1: The acquired $x(t)$ is utilized for the first decomposition by adding a white noise $\omega_i(t)$ with a signal-to-noise ratio β_0 to the original time series $x_i(t)$, as indicated in Equation (1).

$$x_i(t) = x(t) + \beta_0 \omega_i(t) \quad (1)$$

where t stands for the various time points, i for the i th addition of white noise, and n for all the additions of white noise.

Step 2: Use EMD to decompose $x_i(t)$ n times, then obtain $IMF_1^i(t)$. The average value is calculated using Equation (2) to obtain the first IMF of CEEMDAN. The first residual $R_1(t)$ is produced using Equation (3), and $EMD_1(*)$ represents the first IMF obtained through EMD. Theoretically, since white noise has an average value of zero, the influence of white noise can be reduced by finding the average value.

$$\overline{IMF}_1(t) = \frac{1}{n} \sum_{i=1}^n IMF_1^i(t) = \frac{1}{n} EMD_1[x_i(t)] \quad (2)$$

$$R_1(t) = x(t) - \overline{IMF}_1(t) \quad (3)$$

Step 3: The first IMF derived by EMD with the inclusion of white noise $\omega_i(t)$ and signal-to-noise ratio β_1 is the adaptive noise term. The first residual $R_1(t)$ is then combined with the adaptive noise term to create a new time series. The second IMF of CEEMDAN is then obtained by decomposing a fresh time series using Equation (4). Equation (5) is used to generate the second residual $R_2(t)$.

$$\overline{IMF}_2(t) = \frac{1}{n} \sum_{i=1}^n EMD_1\{R_1(t) + \beta_1 EMD_1[\omega_i(t)]\} \quad (4)$$

$$R_2(t) = R_1(t) - \overline{IMF}_2(t) \quad (5)$$

Step 4: Repeat Step 3, adding the new adaptive noise component to the residual term to create the new time series. After that, break it down to get the k th IMF of CEEMDAN. Equations (6) and (7) in particular are as follows:

$$\overline{IMF}_k(t) = \frac{1}{n} \sum_{i=1}^n EMD_1\{R_{k-1}(t) + \beta_{k-1} EMD_{k-1}[\omega_i(t)]\} \quad (6)$$

$$R_k(t) = R_{k-1}(t) - \overline{IMF}_k(t) \quad (7)$$

Step 5: The CEEMDAN algorithm reaches a conclusion when the residual term is unable to proceed with the decomposition since it does not exceed two extreme points. The last residual $R(t)$ at that point is a clear trend term. Equation (8) links the complete IMF to the initial time series of passenger flow.

$$x(t) = \sum_{k=1}^K \overline{IMF_k}(t) + R_k(t) \tag{8}$$

2.2. LSTM Neural Network

Long short-term memory neural network (LSTM) is a special variant of recurrent neural networks (RNN) [34]. The gating mechanism is introduced in comparison to the original RNN, and it may recognize long-term dependencies in the input data. It can address issues like gradient explosion, gradient disappearance, and the difficulty to manage long-term dependencies brought on by intricate network layers. Although URT’s passenger flow significantly varies over the short period, it still depends on changes in both the long-term and current passenger flow levels. Therefore, accurate short-term passenger flow estimates can be made using the LSTM model. Figure 1 depicts the LSTM model structure.

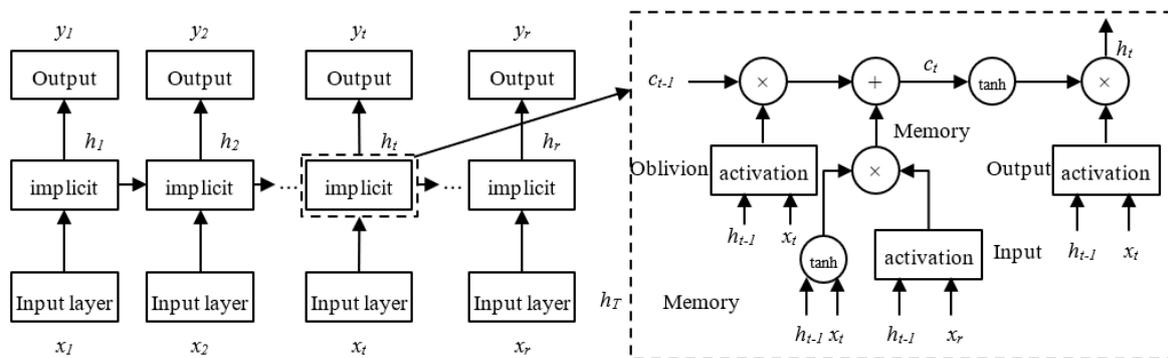


Figure 1. LSTM structure diagram.

The forget gate, shown as f_t in the architectural diagram above, determines whether the upper layer of the LSTM’s hidden cellular state is filtered. i_t stands for the input gate, C_{t-1} for the cell state at the time of the previous moment, C_t for the current moment, and O_t for the output gate. The current input and output are represented by x_t and h_t , respectively. The hyperbolic tangent function is represented by the symbol \tanh , and the sigmoid function is represented by σ . The $w_f, w_i, w_o,$ and w_c stand for the forget gate, input gate, output gate, and weight matrix of the cell state, respectively. The offset vectors for the forget gate, input gate, output gate, and cell state are denoted by $b_f, b_i, b_o,$ and b_c , respectively. Below is a description of each control gate’s calculating principles.

First, the candidate state value \tilde{C} of the input cell at time t and the value of the input gate i_t are calculated:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{9}$$

$$\tilde{C} = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{10}$$

The forget gate’s activation value f_t is then determined at time t :

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{11}$$

It is possible to determine the cell state C_t at time t by using the values discovered in the previous two steps:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{12}$$

The output gate values can be derived after getting the cell state update values:

$$O_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{13}$$

$$h_t = O_t \cdot \tanh(C_t) \tag{14}$$

For the LSTM model selected in this paper, the number of training iterations K , the learning rate L_r , and the number of neurons in the LSTM hidden layer L_1, L_2 , are four hyperparameters that have a significant impact on the algorithm’s performance. The IPSO algorithm is used to adjust and improve the LSTM model, and these four essential hyperparameters are used as features for the particle search.

2.3. PSO Algorithm and Improvement

A swarm intelligence optimization technique called particle swarm optimization (PSO) mimics the social behavior of animals like fish and birds [35]. Velocity and position are the only two characteristics of the particle. Each particle’s position indicates a potential resolution to the issue, and the information that describes it is provided by its position, velocity, and fitness value. Calculating a certain fitness function yields the fitness value.

PSO begins with a set of random particles and uses continual updating and iteration to locate the best solution. Each particle will choose its own position and speed throughout each iteration based on p_b and g_b . Equations (15) and (16) are used to update the particle’s velocity and position after determining these two best values.

$$v_i^{t+1} = \omega v_{iD}^t + c_1 r_1 (p_{bi}^t - x_i^t) + c_2 r_2 (g_{bi}^t - x_i^t) \tag{15}$$

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{16}$$

where v_i is the velocity of the particle; x_i is the particle’s position; c_1 and c_2 are the learning factors; r_1 and r_2 are the random numbers between $[0, 1]$; ω is the inertia weight.

PSO has been successful in many real-world applications, however the standard PSO still struggles with local optimization and has poor convergence accuracy. This study focuses on the three improvement options listed below to address the aforementioned issues.

2.3.1. Improved Adaptive Inertia Weight

The weight of inertia has a major role in determining the convergence of PSO. The local optimization capability is poor but the global capability is higher when the inertia weight is high. The inverse is also accurate. Due to the wide variety of neural network parameters, it is simple to reach a local extremum when using a typical linear decreasing technique, as illustrated in Equation (17). The adaptive change inertia weight, as described in Equation (18), is used in this research to navigate around this restriction.

$$\omega = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{t_{\max}} \times t \tag{17}$$

$$W = 0.1 - \frac{\frac{D}{N \sum g_b / D}}{\sum_{i=1}^N \left[\frac{D}{\sum p_b / D} \right]} \tag{18}$$

where ω_{\max} and ω_{\min} represent the variable’s maximum and minimum values; t and t_{\max} represent the current iteration’s and maximum iteration’s iterations, respectively.

The IPSO algorithm’s early stages are characterized by a modest declining trend, a powerful global search capability, and the potential for a broadly applicable solution. The diminishing trend of W is accelerated in this algorithm’s latter stages. The convergence velocity of IPSO can be accelerated after a good solution is identified in the early stage.

2.3.2. Improvement of Learning Factors

The learning factors c_1 and c_2 are used to regulate the step duration and reposition the particles to reach both the local and the global ideal positions. As the iterative process moves forward in actual applications, it is typically required to adjust the c_1 value from large to tiny in order to speed up the search speed in the initial iterations and enhance the capability of global search. To help with the local refinement search in the subsequent iteration of the iteration and enhance the local search capacity, the c_2 value is changed from small to large. Typically, the PSO algorithm sets $c_1 = c_2 = 2$. However, this falls short of what is required for real-world applications. The linear change learning factors C_1 and C_2 , as shown in Equations (19) and (20), are introduced to improve the global and local search performance of PSO.

$$C_1(t) = 2.5 - 2 \times \frac{t}{t_{\max}} \tag{19}$$

$$C_2(t) = 0.5 + 2 \times \frac{t}{t_{\max}} \tag{20}$$

2.3.3. Improvement of Velocity and Position Update Equation

By inserting a linear model of and as indicated in Equations (21) and (22), the better particle velocity update Equation (23) is created.

$$P_b = \frac{p_b + g_b}{2} \tag{21}$$

$$G_b = \frac{p_b - g_b}{2} \tag{22}$$

$$V_i^{t+1} = WV_i^t + C_1r_1(P_{bi}^t - x_i^t) + C_2r_2(G_{bi}^t - x_i^t) \tag{23}$$

In addition, the average dimensional information conceptual Equation (24) and adaptive determination condition Equation (25) are introduced to further enhance the local and global search capability of particles by adaptively updating the particle positions using “ $X = X + V$ ” and “ $X = WX + (1 - W)V$ ” segments.

$$\delta = \frac{1}{D} \sum_{i=1}^D x_i^t \tag{24}$$

$$Q_i = \frac{\exp\{f[x_i(t)]\}}{\exp\left\{\frac{1}{N} \sum_{i=1}^N f[x_i(t)]\right\}} \tag{25}$$

$$f[x_i(t)] = \frac{1}{n} \sum_{t=1}^n [\hat{x}_i(t) - x_i(t)]^2 \tag{26}$$

$$X_i^{t+1} = \begin{cases} WX_i^t + (1 - W)V_i^{t+1}, & Q_i > \delta \\ X_i^t + V_i^{t+1}, & Q_i < \delta \end{cases} \tag{27}$$

where δ is the average of each particle’s dimensions information; Q_i is the ratio between the current particle’s fitness value and the population’s average fitness value; $f(*)$ is the fitness value of a particle. When $Q_i > \delta$, it implies that IPSO is in the early stages of its search or that the current particle distribution is dispersed, as opposed to the middle or late stages of its search or the concentrated current particle distribution, which are indicated by $Q_i < \delta$.

In summary, the IPSO algorithm finally improves Equations (15) and (16) to Equation (28).

$$\begin{cases} \begin{cases} V_i^{t+1} = WV_i^t + C_1r_1(P_{bi}^t - x_i^t) + C_2r_2(G_{bi}^t - x_i^t) \\ X_i^{t+1} = WX_i^t + (1 - W)V_i^{t+1} \end{cases}, & Q_i > \delta \\ \begin{cases} V_i^{t+1} = WV_i^t + C_1r_1(P_{bi}^t - X_i^t) + C_2r_2(G_{bi}^t - X_i^t) \\ X_i^{t+1} = X_i^t + V_i^{t+1} \end{cases}, & Q_i < \delta \end{cases} \tag{28}$$

2.4. Evaluation Indicators

2.4.1. Benchmark Function

The performance of the proposed IPSO algorithm was evaluated in this study using simulated experiments using the 10 common benchmark functions shown in Table 1 [36]. The prediction model’s convergence precision increases as the test function’s optimized value (f_{opt}) gets nearer to zero.

Table 1. Benchmark functions.

Function	Formulation	Range	f_{opt}
Sphere	$f_1(x) = \sum_{i=1}^D x_i^2$	[-100, 100]	0
Sum Squars	$f_2(x) = \sum_{i=1}^D (ix_i)^2$	[-5.21, 5.21]	0
Sum of Different Power	$f_3(x) = \sum_{i=1}^D x_i ^{i+1}$	[-1, 1]	0
Rosenbrock	$f_4(x) = \sum_{i=1}^D [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	[-30, 30]	0
Quartic	$f_5(x) = \sum_{i=1}^D (ix_i)^2 + random(0,1)$	[-1.28, 1.28]	0
Rastigrin	$f_6(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i) + 10]$	[-5.21, 5.21]	0
Ackley	$f_7(x) = -20 \exp\left(0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D (x_i)^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e$	[-32, 32]	0
Griewank	$f_8(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	[-600, 600]	0
Penalized	$f_9(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^D U(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1), U(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]	0
Penalized2	$f_{10}(x) = \frac{\pi}{D} \left\{ 10 \sin^2(\pi y_1) + \sum_{i=1}^{D-1} (y_i - 1)^2 [1 + 10 \sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\}$ $+ \sum_{i=1}^{D-1} U(x_i, 5, 100, 4), y_i = 1 + \frac{1}{4}(x_i + 1), U(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \leq x_i \leq a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	[-50, 50]	0

2.4.2. Prediction Errors

For evaluating model performance, choosing suitable performance criteria is crucial. All models used in this research are statistically evaluated using the standard deviation (SD), root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), correlation coefficient (R), and coefficient of determination (R^2). The following values would correspond to the projected value and actual value: SD = 0, RMSE = 0, MAE = 0, MAPE = 0, CC = 1, and $R^2 = 1$. The following is a list of the mathematical representations:

$$SD = \sqrt{\frac{1}{n} \sum_{t=1}^n \{ [\hat{y}(t) - \bar{\hat{y}}(t)] - [y(t) - \bar{y}(t)] \}^2} \tag{29}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n [y(t) - x(t)]^2} \tag{30}$$

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y(t) - x(t)| \quad (31)$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{y(t) - x(t)}{x(t)} \right| \quad (32)$$

$$R = \frac{\sum_{t=1}^n [y(t) - \bar{y}(t)][x(t) - \bar{x}(t)]}{\sqrt{\sum_{t=1}^n [y(t) - \bar{y}(t)]^2} \sqrt{\sum_{t=1}^n [x(t) - \bar{x}(t)]^2}} \quad (33)$$

$$R^2 = 1 - \frac{\sum_{t=1}^n [y(t) - x(t)]^2}{\sum_{t=1}^n [\bar{x}(t) - x(t)]^2} \quad (34)$$

where n is the total number of time series samples, $y(t)$ and $x(t)$ are the predicted value and actual value at time t , $\bar{y}(t)$ and $\bar{x}(t)$ are the mean value of the predicted value and actual value.

2.4.3. Taylor Diagram

In addition, this paper further qualitatively evaluates the performance of the prediction models through a Taylor diagram [37]. This diagram can provide a statistical assessment of how well each model matches the other in terms of its SD, RMSE, and R, as well as a simple summary of the degree of connection between simulated and observed fields. The value of R, RMSE, and SD differences between prediction models are all represented by a single point on a two-dimensional plot in a Taylor diagram. Although this diagram's structure is generic, it is particularly helpful when assessing complex models.

2.5. CEEMDAN-IPSO-LSTM Model

The complexity and non-smoothness of the original passenger flow time series of URT interfere with the neural network prediction and the problems of neural network hyperparameters determined by trial-and-error with only empirical values seriously affecting the accuracy of the prediction model. In this study, we use the CEEMDAN algorithm to break down the time series data for the passenger flow, use the LSTM hyperparameters as the object of optimization, combine them with the IPSO algorithm to determine the optimal value of the LSTM hyperparameters, and build a combined CEEMDAN-IPSO-LSTM model to accurately predict the short-term passenger flow of URT systems. Figure 2 depicts the precise prediction method, and the subsequent steps are presented in the prediction process.

Step 1: Data decomposition. CEEMDAN is used to decompose passenger flow data to obtain IMFs and Res.

Step 2: A training set and a test set are created from the passenger flow sequence that was obtained from CEEMDAN decomposition.

Step 3: Construct LSTM neural network. Initialize the batch size, hidden layer unit number, gradient limit, and other parameters of LSTM.

Step 4: Initialize the IPSO parameters at random. The size of the population, the maximum number of iterations, and the size of the particles are chosen at random.

Step 5: Create the CEEMDAN-IPSO-LSTM prediction model and build a combination prediction model; the hyperparameters (L_1, L_2, L_r, K) of LSTM are computed using IPSO. If the iteration termination conditions are met, output the optimal value of LSTM hyperparameters. If it is not satisfied, make $t = t + 1$, and repeat steps 2-5.

Step 6: Evaluate the prediction model. CEEMDAN-IPSO-LSTM model is evaluated by the prediction error and Taylor diagram.

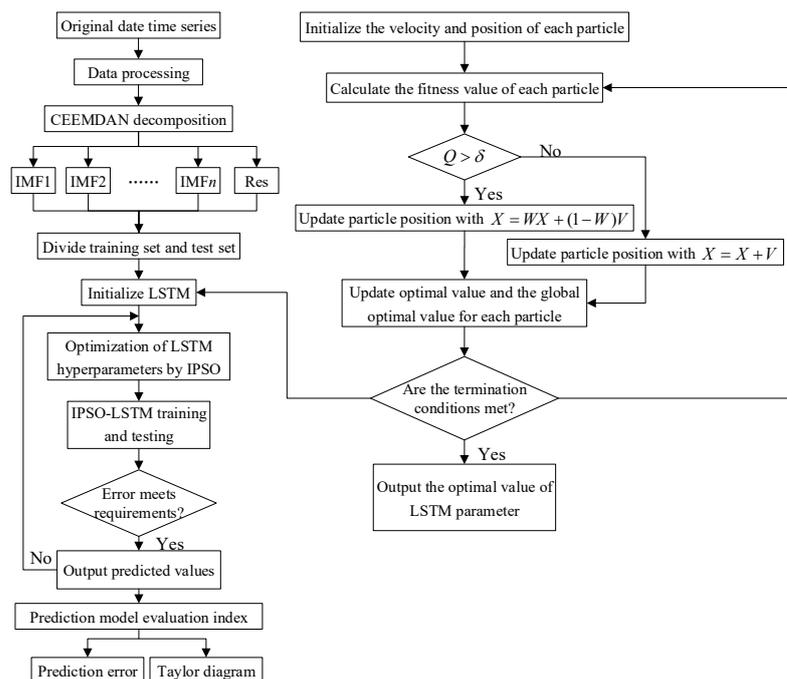


Figure 2. Flowchart of CEEMDAN-IPSO-LSTM prediction model.

3. Results

3.1. Data Set

The experimental data are the inbound and outbound passenger flow data of Yangji Station of Guangzhou Metro from 1 July 2019 to 28 July 2019 from 6:15 to 23:15. The time series was smoothed by aggregating flow data into nonoverlapping 15-min intervals [38]. This resulted in 96 samples per day. Based on the above CEEMDAN-IPSO-LSTM model, the first 75% of the data were taken as the training set and the last 25% as the test set. The sliding window length was 3; that is, the data of the first 3 weeks were used to predict the next week.

Figure 3 depicts how Yangji Station’s inbound/outbound passenger flow statistics changed throughout the experiment. Additionally, because the subway station is close to sizable residential neighborhoods, commuters frequently utilize it during the working week, and significant morning and evening peak characteristics exist, which aids in improving forecast performance. The passenger flow significantly varies during the course of a single day, as shown in Figure 3. Its pattern is quite similar during the working week, with two peaks visible each day. The first inbound/outbound peak typically occurs between 7:30 and 8:45 and 7:30 and 9:30 in the morning, and the second inbound/outbound peak usually occurs between 17:15 and 19:15 and 17:45 and 19:00 in the afternoon. The passenger volume during the morning and/or afternoon peaks is often two to three times more than during off-peak times. Weekend trends diverge from weekday trends, and there are no clear morning and afternoon peaks. Between 11:00 and 19:00, there are frequently high passenger loads. In general, Saturday has a greater passenger volume than Sunday. Due to entertainment and social events, it is also observed that there is an increase in passenger traffic late on Friday and Saturday nights.

3.2. CEEMDAN Decomposition

The inbound passenger flow time series was divided using CEEMDAN into a total of 12 subseries with various amplitudes and frequencies, comprising 11 IMF components and a Res component, as shown in Figure 4. It is clear that when the IMF is further decomposed, it becomes less volatile and cyclical, which is consistent with the decomposed IMF’s features. IMF1 has the highest frequency and the shortest wavelength. As the wavelength rises, the frequency of IMF2 to IMF11 drops in turn. The trend term of the inbound passenger flow sequence is represented by the residual term.

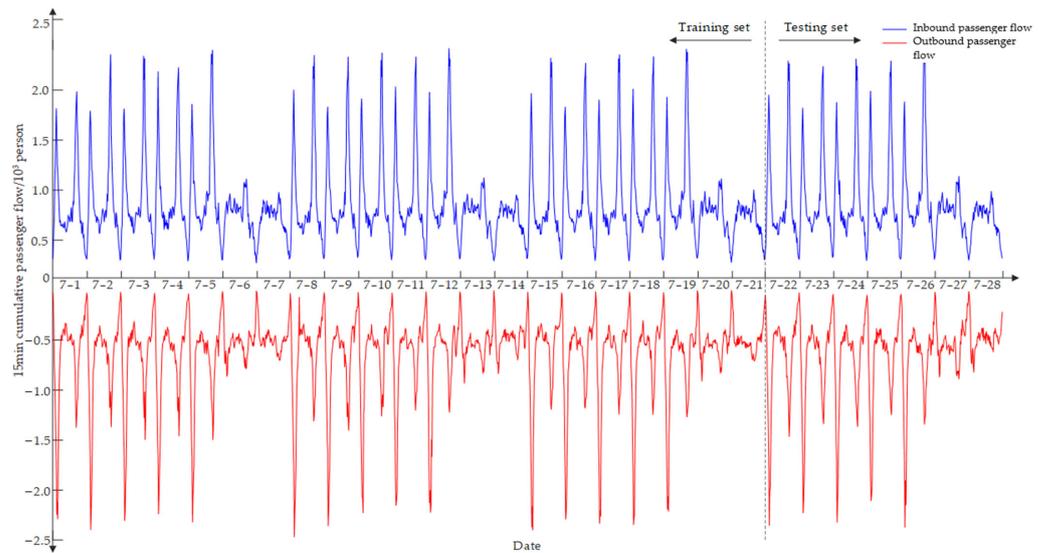


Figure 3. Four weeks of inbound and outbound passenger flow data.

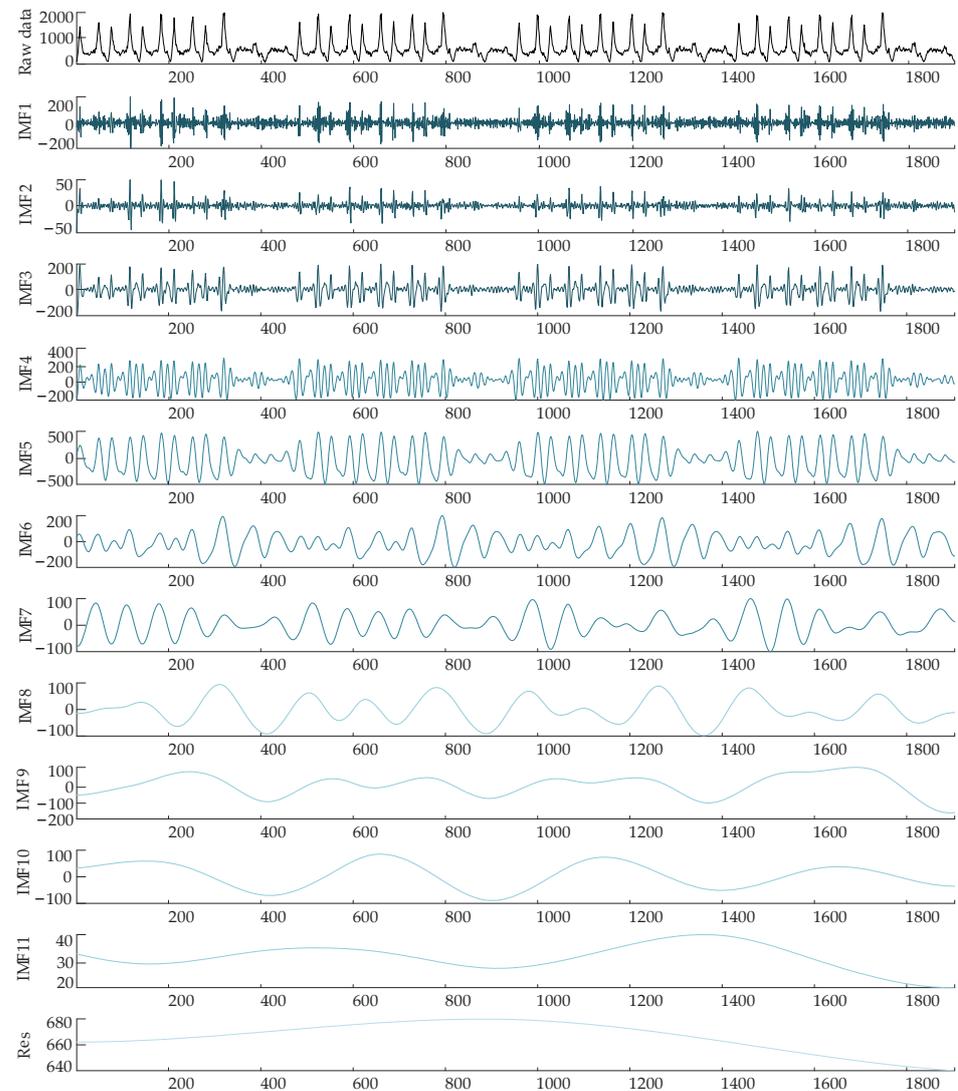


Figure 4. IMFs and Res obtained from the daily inbound passenger flow data after CEEMDAN decomposition.

3.3. Benchmark Function and Comparison Algorithm

Four other evolutionary algorithms (SOA [39], WOA [40], GWO [41], and PSO) were chosen for comparison with IPSO to assess the IPSO algorithm’s performance. All comparison algorithms made use of the same set of parameters to ensure fairness. The maximum number of iterations was 1000, and the population size was set at 50. Additionally, each algorithm was individually run 50 times on each benchmark function to lessen the effect of random numbers on algorithm performance.

Table 2 compares five evolutionary algorithms across ten benchmark functions. The operation results in Table 2 show that, for the identical benchmark function, the IPSO algorithm’s minimum, maximum, mean, and SD values are, for the most part, smaller than those of other algorithms. It can be seen from the operation results in Table 2 that, under the same benchmark function, the value of minimum, maximum, mean, and SD obtained by the IPSO algorithm are smaller than other algorithms, in most cases. The IPSO algorithm performs better than other algorithms in the whole iteration process, which can enable particles to gather more stably near the global optimal value and more easily find the global optimal solution.

Table 2. Comparison results between IPSO and other evolutionary algorithms.

Function	Value	SOA	WOA	GWO	PSO	IPSO
f_1	Min	3.43×10^{-3}	4.59×10^{-13}	5.38×10^{-21}	1.12×10	1.83×10^{-35}
	Max	6.33×10	3.78×10^{-12}	1.96×10^{-17}	1.79×10	1.51×10^{-32}
	Mean	5.21×10^0	1.54×10^{-12}	1.49×10^{-18}	1.49×10	2.70×10^{-33}
	Std	1.26×10	8.83×10^{-13}	3.71×10^{-18}	1.46×10^0	3.62×10^{-33}
	Rank	5	3	2	4	1
f_2	Min	3.43×10^{-3}	3.14×10^{-10}	5.38×10^{-13}	1.42×10	6.06×10^{-21}
	Max	6.33×10	1.30×10^{-9}	2.88×10^{-9}	1.78×10	3.11×10^{-19}
	Mean	5.21×10^0	6.15×10^{-10}	4.56×10^{-10}	1.59×10	7.02×10^{-20}
	Std	1.26×10	2.19×10^{-10}	7.18×10^{-10}	9.45×10^{-1}	6.26×10^{-20}
	Rank	4	3	2	5	1
f_3	Min	1.59×10^3	1.15×10^4	4.11×10^{-18}	4.23×10	2.09×10^{-11}
	Max	1.40×10^3	2.34×10^4	1.73×10^{-12}	8.92×10	7.04×10^{-7}
	Mean	6.69×10^3	1.62×10^4	6.59×10^{-14}	6.82×10	6.57×10^{-8}
	Std	3.65×10^3	2.85×10^4	2.19×10^{-10}	1.15×10	1.43×10^{-7}
	Rank	4	5	1	3	2
f_4	Min	9.74×10^0	1.25×10	1.41×10^{-11}	1.31×10^0	1.61×10^{-9}
	Max	3.96×10	2.68×10	7.01×10^{-9}	1.71×10^0	6.49×10^{-8}
	Mean	2.32×10	2.00×10	1.37×10^{-9}	1.55×10^0	1.83×10^{-8}
	Std	9.64×10^0	3.63×10^0	1.68×10^{-9}	9.59×10^{-2}	1.53×10^{-8}
	Rank	5	4	2	3	1
f_5	Min	1.00×10^{-4}	9.19×10	1.43×10	1.76×10^2	0.00×10^0
	Max	1.26×10^2	1.87×10^2	1.44×10^2	2.57×10^2	2.07×10^0
	Mean	3.01×10	1.46×10^2	6.34×10	2.21×10^2	6.91×10^{-2}
	Std	2.99×10	2.64×10	3.09×10	1.56×10	3.79×10^{-1}
	Rank	5	3	2	4	1
f_6	Min	7.18×10	2.47×10	2.85×10	1.66×10^3	2.54×10
	Max	5.02×10^4	4.56×10	2.87×10	5.20×10^3	2.79×10
	Mean	7.00×10^3	2.72×10	2.87×10	3.20×10^3	2.66×10
	Std	1.21×10^4	3.58×10^0	3.20×10^{-2}	9.90×10^2	7.18×10^{-1}
	Rank	5	4	3	2	1
f_7	Min	4.29×10^0	3.66×10^{-13}	1.57×10^{-2}	1.07×10	2.05×10^{-5}
	Max	1.79×10	2.98×10^{-12}	1.39×10^{-1}	1.81×10	1.01×10^0
	Mean	7.75×10^0	1.52×10^{-12}	6.56×10^{-2}	1.53×10	4.64×10^{-1}
	Std	3.34×10^0	7.21×10^{-13}	3.03×10^{-2}	1.99×10^0	3.24×10^{-1}
	Rank	5	4	3	2	1
f_8	Min	5.24×10^0	3.34×10^{-2}	2.46×10^{-5}	1.62×10^2	2.37×10^{-4}
	Max	2.89×10^5	8.20×10^{-2}	7.11×10^{-4}	4.18×10^2	2.62×10^{-3}
	Mean	4.69×10^4	5.33×10^{-2}	2.40×10^{-4}	2.65×10^2	1.31×10^{-3}
	Std	7.42×10^4	1.40×10^{-2}	1.86×10^{-4}	6.37×10	5.97×10^{-4}
	Rank	5	4	3	2	1

Table 2. Cont.

Function	Value	SOA	WOA	GWO	PSO	IPSO
f_9	Min	-1.12×10^3	-1.66×10^3	-1.30×10^3	-1.52×10^3	-1.49×10^3
	Max	-7.88×10^2	-1.33×10^3	-9.10×10^2	-9.46×10^2	-1.08×10^3
	Mean	-9.34×10^2	-9.34×10^2	-1.09×10^3	-1.19×10^4	-1.24×10^3
	Std	7.27×10	7.97×10	1.03×10^2	1.47×10^2	8.49×10
	Rank	2	1	5	3	4
f_{10}	Min	3.34×10^{-1}	1.00×10^{-22}	5.00×10^{-4}	2.02×10^{-1}	0.00×10^0
	Max	2.53×10	2.31×10^{-19}	4.85×10^{-1}	7.98×10^0	5.23×10^{-2}
	Mean	2.30×10^0	2.76×10^{-20}	6.85×10^{-2}	1.37×10^0	2.18×10^{-2}
	Std	5.49×10^0	5.08×10^{-20}	1.41×10^{-1}	1.81×10^0	1.26×10^{-2}
	Rank	5	4	2	3	1
Total Rank		45	35	25	31	14
Final Rank		5	4	2	3	1

Figure 5 displays the ideal iterative convergence curves for each benchmark function. The convergence curve of the IPSO algorithm on most benchmark functions is below that of other algorithms. It demonstrates that IPSO not only has great convergence accuracy throughout the whole search process for each specified benchmark function, but also a faster convergence speed. The IPSO algorithm’s adaptive strategy significantly enhances the efficiency of particle optimization, avoids PSO’s inefficient iteration process, and achieves a balance between local and global search.

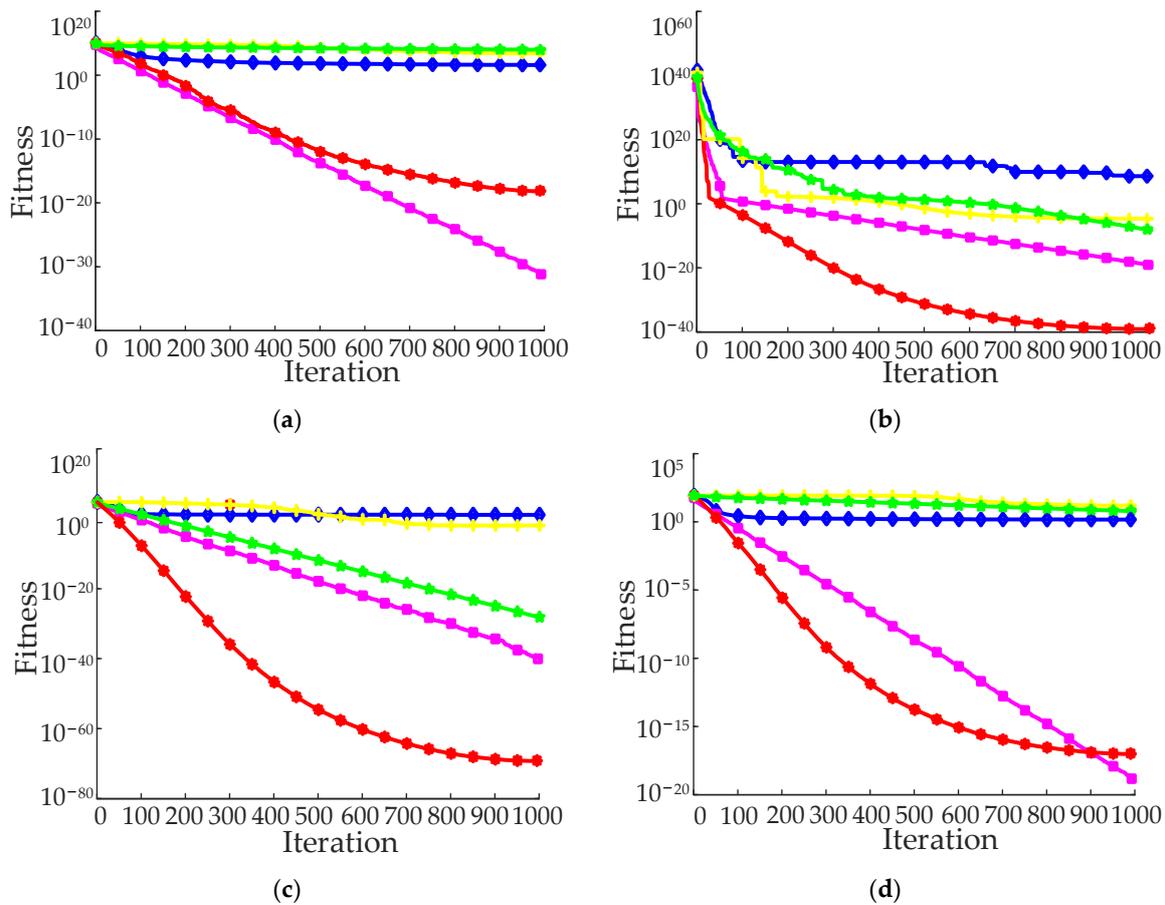


Figure 5. Cont.

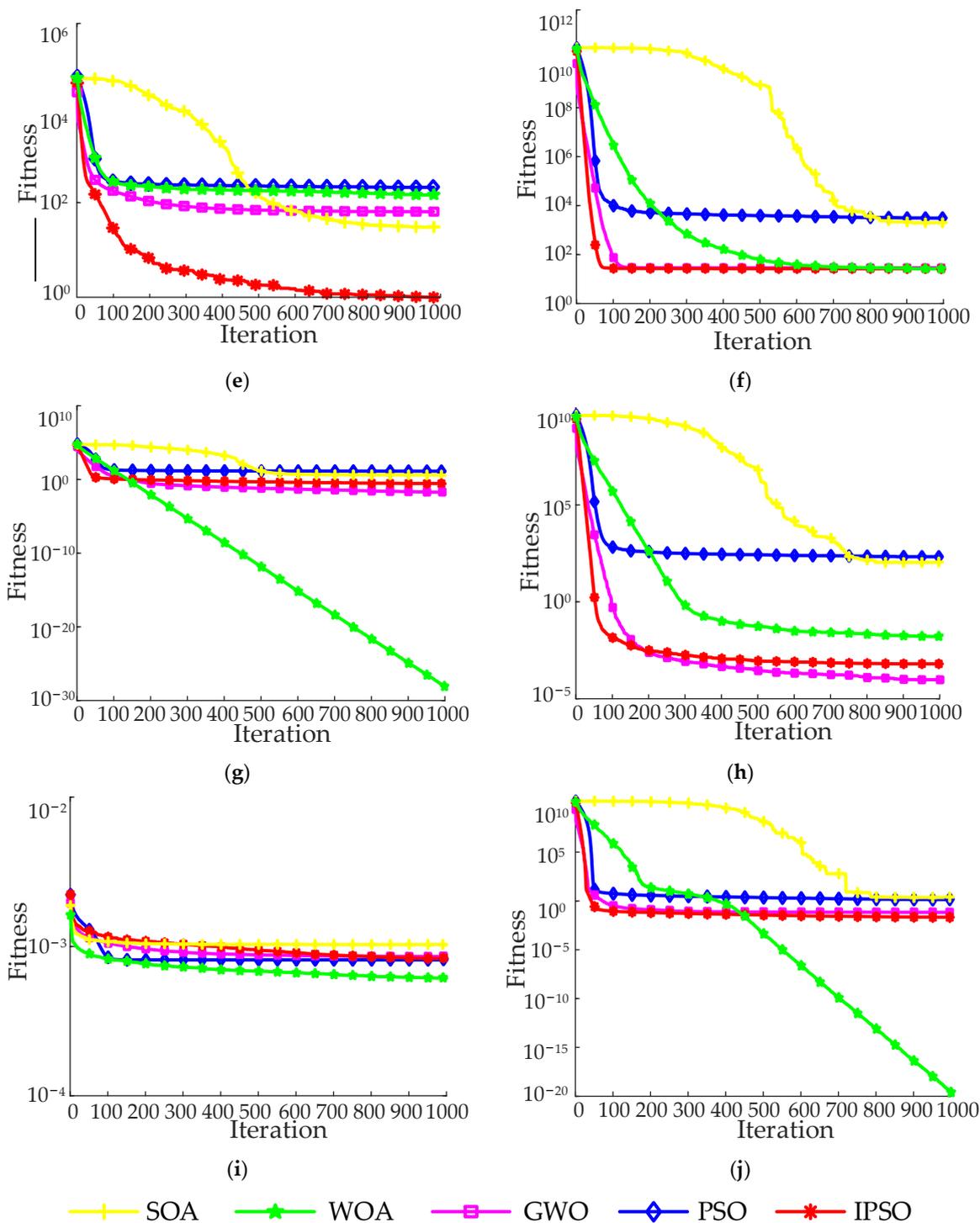


Figure 5. Average convergence curves of 10 benchmark functions. (a–j) represent the average convergence curve of function f_1 – f_{10} .

3.4. CEEMDAN-IPSO-LSTM Results

The fitness function employed in this study is the best mean square error (MSE) that the LSTM could attain throughout training. The hyperparameters derived from the optimization are L_1 , L_2 , L_r , and K , which correspond to the minimum MSE. Figure 6a depicts the error convergent curve during the training process. It was discovered that as the iteration count increased, the error of the CEEMDAN-IPSO-LSTM model soon converged. Within four iterations, the CEEMDAN-IPSO-LSTM fitness evolution curve attained the necessary

precision and then maintained the ideal fitness value, demonstrating strong learning ability. The initial and final errors of CEEMDAN-IPSO-LSTM are one order of magnitude fewer than those of CEEMDAN-PSO-LSTM, and the model accuracy significantly increases. Figure 6b displays the estimated outcomes of the LSTM hyperparameters, which are $L_1 = 65, L_2 = 173, L_r = 0.007,$ and $K = 60,$ which were optimized by PSO and IPSO.

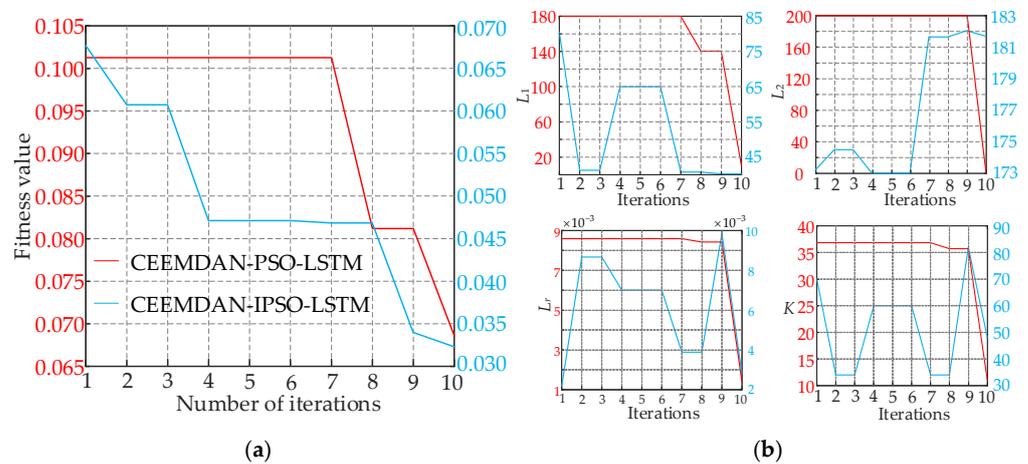


Figure 6. (a) The error convergence curves of CEEMDAN-PSO-LSTM model and CEEMDAN-IPSO-LSTM model in the training process. (b) The hyperparameters optimization results of CEEMDAN-PSO-LSTM model and CEEMDAN-IPSO-LSTM model.

3.5. Prediction Results of Inbound and Outbound Passenger Flow

The LSTM, CEEMDAN-LSTM, and CEEMDAN-PSO-LSTM models were employed for comparison testing to confirm the accuracy of the proposed CEEMDAN-IPSO-LSTM model. Figure 7 displays the outcomes of several model predictions of data on the inbound and outgoing passenger flow. As can be observed, the trend of the actual value curves, whether during the peak time or off-peak period, is largely consistent with the forecast curves derived by various models. The CEEMDAN-IPSO-LSTM model, on the other hand, correlates to a prediction curve through thorough local observation, which has greater forecast accuracy than the other models and is more similar to the real monitoring curve, indicating the CEEMDAN-IPSO-LSTM model has strong robustness.

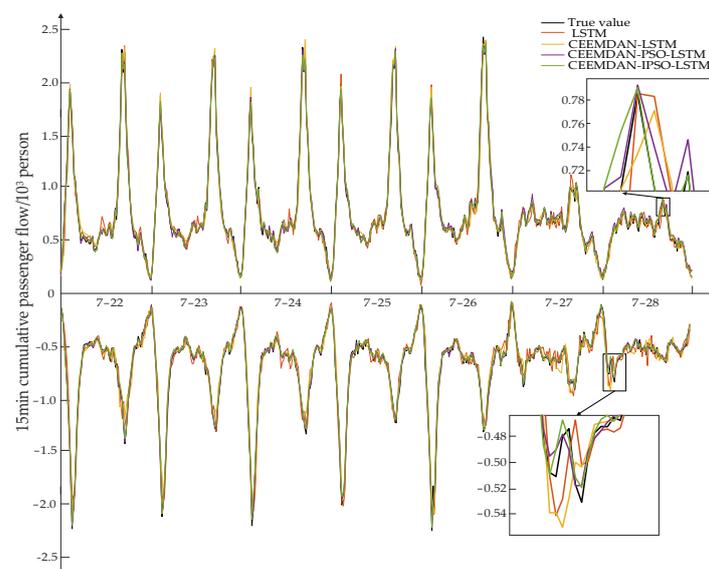


Figure 7. Prediction results of inbound and outbound passenger flow of different models in the last week.

3.6. Evaluation Indicators of Prediction Models

3.6.1. Quantitative Analysis Based on Prediction Errors

Table 3 shows the performance of the CEEMDAN-IPSO-LSTM model comparison to other models (LSTM, CEEMDAN-LSTM, CEEMDAN-PSO-LSTM) for both inbound and outbound passenger flow data. It can be seen that the CEEMDAN-IPSO-LSTM model respectively reduces SD, RMSE, MAE, and MAPE of inbound/outbound passenger flow data concerning the whole day of month by 12~40 persons/13~35 persons, 13~44 person/12~35 persons, 6~37 persons/12~31 persons and 5.08~46.89%/6.5~35.1%, R and R² respectively increased by 0.07~2.32%/0.86~3.63% and 0.13~2.19%/0.67~1.67%. At the same time, the proposed model can achieve favorable prediction results for the different periods during weekdays and also on the weekend. This demonstrates once more the higher prediction accuracy of the CEEMDAN-IPSO-LSTM model suggested in this study.

Table 3. Comparison of prediction errors.

Period	Error	Inbound				Outbound				
		L ¹	C-L ¹	C-P-L ¹	C-IP-L ¹	L ¹	C-L ¹	C-P-L ¹	C-IP-L ¹	
Month	Day	SD	67	55	39	27	89	83	67	54
		RMSE	69	55	38	25	89	84	66	54
		MAE	57	41	26	20	66	60	47	35
		MAPE	82.36	73.19	40.55	35.47	83.68	80.11	55.08	48.58
		R	97.24	98.15	99.49	99.56	95.16	96.65	97.93	98.79
		R ²	97.69	99.10	99.75	99.88	97.57	97.87	98.57	99.24
	Day	SD	62	49	33	25	86	78	45	37
		RMSE	61	47	33	24	87	78	45	37
		MAE	44	36	24	19	63	56	42	36
		MAPE	73.34	65.84	40.02	35.33	72.04	70.53	57.82	40.68
		R	98.81	99.17	99.49	99.70	96.34	97.04	98.02	99.33
		R ²	98.34	99.42	99.74	99.86	98.21	98.57	99.52	99.67
Weekday	Peak	SD	124	112	86	78	199	151	124	100
		RMSE	123	110	87	75	203	151	126	101
		MAE	91	83	67	52	159	112	91	83
		MAPE	63.17	58.84	47.37	32.01	61.77	57.84	47.02	41.63
		R	88.33	94.54	96.24	98.7	74.20	80.66	90.12	94.73
		R ²	93.98	94.73	97.53	98.35	87.01	93.40	95.11	97.24
	Off-Peak	SD	43	41	32	26	69	59	41	32
		RMSE	43	40	33	27	70	57	41	32
		MAE	33	29	23	20	50	43	30	25
		MAPE	72.82	62.72	50.49	47.73	70.25	68.56	56.14	52.08
		R	95.11	95.80	97.28	96.79	87.98	91.77	95.73	97.39
		R ²	97.54	97.92	98.70	99.43	93.91	95.80	97.97	98.69
Weekend	Day	SD	116	92	60	41	120	99	73	52
		RMSE	119	93	60	42	118	100	73	55
		MAE	87	71	50	46	89	75	50	42
		MAPE	37.28	26.27	18.71	15.00	43.31	40.19	35.71	32.00
		R	79.88	83.48	88.37	92.84	73.88	80.48	89.35	91.84
		R ²	84.06	89.64	93.18	96.88	86.29	90.64	94.18	97.59

¹ The names of LSTM, CEEMDAN-LSTM, CEEMDAN-PSO-LSTM, and CEEMDAN-IPSO-LSTM models are abbreviated as L, C-L, C-P-L, and C-IP-L in Table 3.

3.6.2. Qualitative Analysis Based on Taylor Diagram

Additionally, a Taylor diagram was created for each model's prediction errors in order to qualitatively assess the characteristics of how prediction errors are distributed among different prediction models. According to Figure 8, the comprehensive ranking of prediction results is as follows: LSTM < EMD-LSTM < EEMD-LSTM < CEEMD-LSTM < CEEMDAN-LSTM < EMD-PSO-LSTM < EEMD-PSO-LSTM < CEEMD-PSO-LSTM <

CEEMDAN-IPSO-LSTM < EMD-IPSO-LSTM < EEMD-IPSO-LSTM < CEEMD-IPSO-LSTM < CEEMDAN-IPSO-LSTM. Among the peer models, the CEEMDAN-IPSO-LSTM model has the highest accuracy and can meet the demands for accurate short-term predictions of passenger flow.

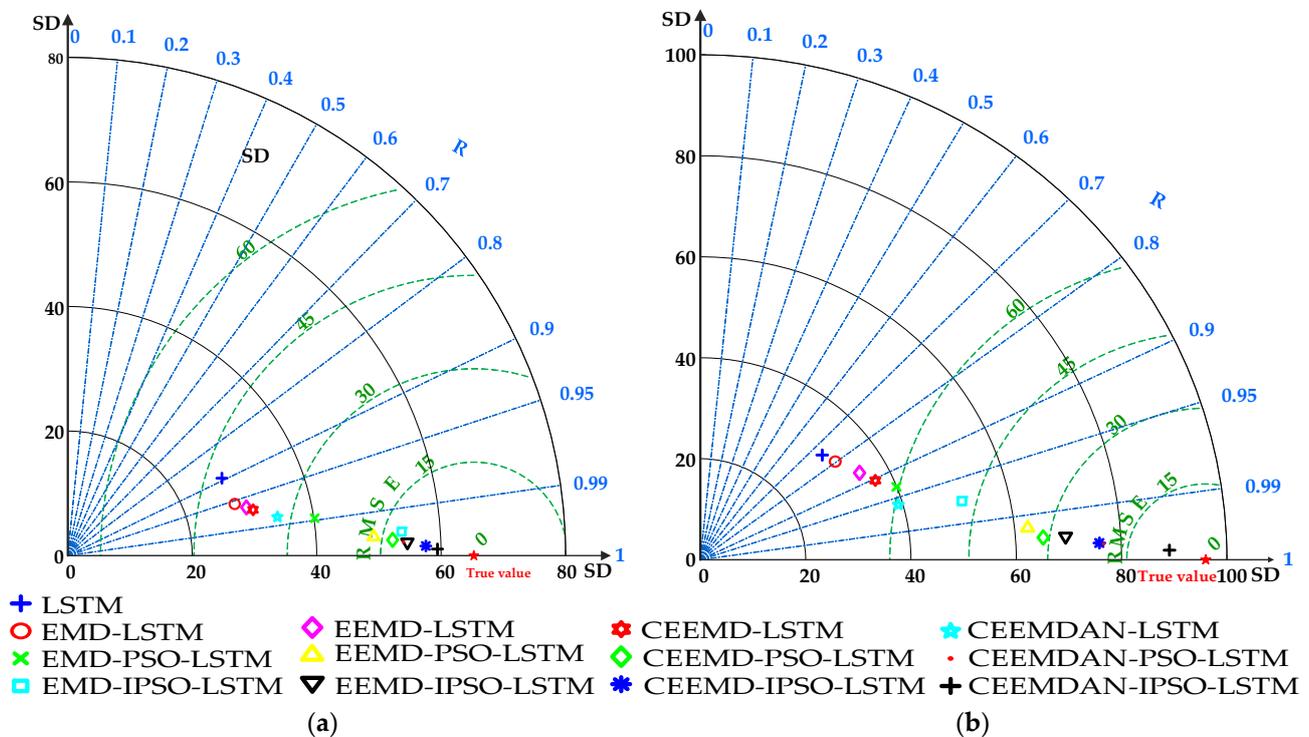


Figure 8. Taylor diagram of 13 prediction models. (a) represents the inbound passenger flow, (b) represents the outbound passenger flow.

4. Discussion

In this paper, we verified that the CEEMDAN-IPSO-LSTM model can accurately predict short-term passenger flow of URT. The error statistics of inbound passenger flow and outbound passenger flow demonstrate that the proposed model, combining the strong noise-resistant robustness of the CEEMDAN and the nonlinear mapping of the LSTM, outperforms other models in terms of prediction performance. Compared with the single LSTM model, the CEEMDAN-IPSO-LSTM model reduce by 40 person/35 person, 44 person/35 person, 37 person/31 person, and 46.89%/35.1% in SD, RMSE, MAE, and MAPE, and increase by 2.32%/3.63% and 2.19%/1.67% in R and R², respectively. The performance improvement of CEEMDAN-IPSO-LSTM for the LSTM is significantly higher than that of the other models.

Because of the sensitivity of the short-term prediction model to the original passenger flow time series, it can consider the impact of various factors on the passenger flow series. For further study, more effective pretreatment methods of noise reduction for passenger flow data should be explored and applied to further enhance the algorithm performance. The methods that could be explored include variational mode decomposition [42], synchrosqueezing wavelet transform [43], savitzky-golay filter [44], etc.

In this paper, we only analyzed a basic prediction model of LSTM. There exist some other improvements to this model. For example, the Bi-directional LSTM [45] and gated recurrent neural network [46]. Therefore, more base models with various denoising methods should be compared and analyzed, to further strengthen the applicability of the IPSO-LSTM model in passenger flow prediction.

In addition, the CEEMDAN-IPSO-LSTM model proposed in this paper is also valuable for time series prediction of other traffic flows. At the same time, the model can be further

extended from one subway station to one subway line, or even to the entire subway network, to improve the accurate prediction of short-term passenger flow in the URT system.

5. Conclusions

There are increasing traffic pollution issues in the process of urbanization in many countries. URT is low-carbon and widely regarded as an effective way to solve such problems. The accurate prediction of short-term passenger flow in URT systems can improve the efficiency of transport infrastructure and vehicles, and provide reference for the development of low-carbon transportation. In this study, a short-term passenger flow prediction model for URT was proposed based on CEEMDAN-IPSO-LSTM, including the framework design of CEEMDAN-IPSO-LSTM and the determination of model parameters, which successfully addresses the issues of easy local optimum fall-off, slow late convergence, and early convergence in the conventional PSO algorithm. The experimental findings showed that the CEEMDAN-IPSO-LSTM model beat other comparison models in terms of overall performance. Specifically, the CEEMDAN-IPSO-LSTM model respectively reduced SD, RMSE, MAE, and MAPE of inbound/outbound passenger flow data concerning the whole day of month by 12~40 person/13~35 person, 13~44 person/12~35 person, 6~37 person/12~31 person and 5.08~46.89%/6.5~35.1%, R and R² respectively increased by 0.07~2.32%/0.86~3.63% and 0.13~2.19%/0.67~1.67%. At the same time, the proposed model achieved favorable prediction results during weekdays and at the weekend. In summary, this research validates the applicability and robustness of the CEEMDAN-IPSO-LSTM model in the area of predicting short-term passenger flow for URT systems, and extends the use of ensemble learning technology.

However, there are still a number of restrictions in this study. For instance, the current case study examined the station's passenger flow statistics, but did not address the relationships between other lines, nor did investigate how service interruptions and spatiotemporal impacts can affect passenger flow. Additionally, multi-source data pertaining to factors such as weather, traffic, and accidents might be investigated in the future. Further research into the proposed model's applicability to other spatial-temporal data mining applications, such trajectory prediction, would also be interesting.

Author Contributions: Conceptualization, L.Z. and Z.L.; methodology, L.Z. and Z.L.; software, L.Z. and Z.L.; validation, L.Z. and Z.L.; formal analysis, L.Z. and Z.L.; investigation, L.Z. and Z.L.; resources, L.Z., J.Y. and X.X.; data curation, L.Z. and Z.L.; writing—original draft preparation, L.Z. and Z.L.; writing—review and editing, L.Z., Z.L., J.Y. and X.X.; visualization, L.Z., Z.L.; supervision, L.Z., Z.L., J.Y. and X.X.; project administration, L.Z., Z.L., J.Y. and X.X.; funding acquisition, L.Z., J.Y. and X.X. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China (No. 62063009); the State Key Laboratory of Rail Traffic Control and Safety (Contract No. RCS2020K005), Beijing Jiaotong University; the Science and Technology Project of the Education Department of Jiangxi Province (No.GJJ200825); Scientific research project of Ganjiang Innovation Academy, Chinese Academy of Sciences (No.E255J001); and Jiangxi University of Scientific and Technology research fund for high-level talents (No.205200100428).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used and/or analyzed during the current study are available from the corresponding author upon reasonable request.

Acknowledgments: The authors would like to sincerely thank the editors and the anonymous reviewers for their constructive comments that greatly contributed to improving the manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Xi, J. *Statement by H.E. Xi Jinping President of the People's Republic of China at the General Debate of the 75th Session of the United Nations General Assembly*; Ministry of Foreign Affairs, the People's Republic of China: Beijing, China, 2020.
2. Zhang, Y.; Jiang, L.; Shi, W. Exploring the growth-adjusted energy-emission efficiency of transportation industry in China. *Energy Econ.* **2020**, *90*, 104873. [CrossRef]
3. Mao, R.; Bao, Y.; Duan, H.; Liu, G. Global urban subway development, construction material stocks, and embodied carbon emissions. *Humanit. Soc. Sci. Commun.* **2021**, *8*, 83. [CrossRef]
4. Wei, T.; Chen, S. Dynamic energy and carbon footprints of urban transportation infrastructures: Differentiating between existing and newly-built assets. *Appl. Energy* **2020**, *277*, 115554. [CrossRef]
5. China Association of Metros. Available online: <https://www.camet.org.cn/xxfb> (accessed on 18 October 2022).
6. Liu, Y.; Liu, Z.; Jia, R. DeepPF: A deep learning based architecture for metro passenger flow prediction. *Transp. Res. Part C Emerg. Technol.* **2019**, *101*, 18–34. [CrossRef]
7. Li, G.; Knoop, V.L.; van Lint, H. Multistep traffic forecasting by dynamic graph convolution: Interpretations of real-time spatial correlations. *Transp. Res. Part C Emerg. Technol.* **2021**, *128*, 103185. [CrossRef]
8. Cheng, Z.; Trépanier, M.; Sun, L. Incorporating travel behavior regularity into passenger flow forecasting. *Transp. Res. Part C Emerg. Technol.* **2021**, *128*, 103200. [CrossRef]
9. Noursalehi, P.; Koutsopoulos, H.N.; Zhao, J. Predictive decision support platform and its application in crowding prediction and passenger information generation. *Transp. Res. Part C Emerg. Technol.* **2021**, *129*, 103139. [CrossRef]
10. Jiao, P.; Li, R.; Sun, T.; Hou, Z.; Ibrahim, A. Three Revised Kalman Filtering Models for Short-Term Rail Transit Passenger Flow Prediction. *Math. Probl. Eng.* **2016**, *2016*, 9717582. [CrossRef]
11. Liang, S.; Ma, M.; He, S.; Zhang, H. Short-Term Passenger Flow Prediction in Urban Public Transport: Kalman Filtering Combined K-Nearest Neighbor Approach. *IEEE Access* **2019**, *7*, 120937–120949. [CrossRef]
12. Cao, L.; Liu, S.G.; Zeng, X.H.; He, P.; Yuan, Y. Passenger Flow Prediction Based on Particle Filter Optimization. *Appl. Mech. Mater.* **2013**, *373–375*, 1256–1260. [CrossRef]
13. Liu, S.Y.; Liu, S.; Tian, Y.; Sun, Q.L.; Tang, Y.Y. Research on Forecast of Rail Traffic Flow Based on ARIMA Model. *J. Phys. Conf. Ser.* **2021**, *1792*, 012065. [CrossRef]
14. Guo, J.; Huang, W.; Williams, B.M. Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 50–64. [CrossRef]
15. Shahriari, S.; Ghasri, M.; Sisson, S.A.; Rashidi, T. Ensemble of ARIMA: Combining parametric and bootstrapping technique for traffic flow prediction. *Transp. A Transp. Sci.* **2020**, *16*, 1552–1573. [CrossRef]
16. Hu, Y.; Wu, C.; Liu, H. Prediction of passenger flow on the highway based on the least square support vector machine. *Transport* **2011**, *26*, 197–203. [CrossRef]
17. Zhou, G.; Tang, J. Forecast of Urban Rail Transit Passenger Flow in Holidays Based on Support Vector Machine Model. In Proceedings of the 5th International Conference on Electromechanical Control Technology and Transportation (ICECTT), Nanchang, China, 15–17 May 2020.
18. Li, H.; Zhang, J.; Yang, L.; Qia, J.; Gaoa, Z. Graph-GAN: A spatial-temporal neural network for short-term passenger flow prediction in urban rail transit systems. *Transp. Res. Part C* **2022**, 1–24.
19. Zhang, J.; Chen, F.; Cui, Z.; Guo, Y.; Zhu, Y. Deep Learning Architecture for Short-Term Passenger Flow Forecasting in Urban Rail Transit. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7004–7014. [CrossRef]
20. Yu, S.; Shang, C.; Yu, Y.; Zhang, S.; Yu, W. Prediction of bus passenger trip flow based on artificial neural network. *Adv. Mech. Eng.* **2016**, *8*, 1–7. [CrossRef]
21. Long, X.; Li, J.; Chen, Y. Metro short-term traffic flow prediction with deep learning. *Control. Decis.* **2019**, *34*, 1589–1600.
22. Nicholas, G.; Sokolov, V. Deep learning for short-term traffic flow prediction. *Transp. Res. Part C Emerg. Technol.* **2017**, *79*, 1–17.
23. Zhao, Z.; Chen, W.; Wu, X. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2019**, *13*, 68–75. [CrossRef]
24. He, P.; Jiang, G.; Lam, S.; Sun, Y. Learning heterogeneous traffic patterns for travel time prediction of bus journeys. *Inf. Sci.* **2020**, *512*, 1394–1406. [CrossRef]
25. Jing, Y.; Hu, H.; Guo, S.; Wang, X.; Chen, F. Short-Term Prediction of Urban Rail Transit Passenger Flow in External Passenger Transport Hub Based on LSTM-LGB-DRS. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 4611–4621. [CrossRef]
26. Gong, M.; Fei, X.; Wang, Z.H.; Qiu, Y.J. Sequential Framework for Short-Term Passenger Flow Prediction at Bus Stop. *Transp. Res. Rec. J. Transp. Res. Board* **2014**, *2417*, 58–66. [CrossRef]
27. Lan, Q.; Weide, L.; Shijia, L. Effective passenger flow forecasting using STL and ESN based on two improvement strategies. *Neurocomputing* **2019**, *356*, 244–256.
28. Guo, J.; Xie, Z.; Qin, Y.; Jia, L.; Wang, Y. Short-term abnormal passenger flow prediction based on the fusion of SVR and LSTM. *IEEE Access* **2019**, *7*, 42946–42955. [CrossRef]
29. Liu, L.; Chen, R. A novel passenger flow prediction model using deep learning methods. *Transp. Res. Part C* **2017**, *84*, 74–91. [CrossRef]
30. Marfa, E.; Torres, M.; Colominas, G.; Patrick, F. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the IEEE International Conference on Acoustics, Prague, Czech Republic, 22–27 May 2011.

31. Huang, N.E.; Shen, Z.; Long, S.R.; Wu, M.C.; Shih, H.H.; Zheng, Q.; Yen, N.-C.; Tung, C.C.; Liu, H.H. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proc. R. Soc. Lond. Ser. A Math. Phys. Eng. Sci.* **1998**, *454*, 903–995. [[CrossRef](#)]
32. Wu, Z.; Huang, N.E. Ensemble empirical mode decomposition: A noise-assisted data analysis method. *Adv. Adapt. Data Anal.* **2009**, *1*, 1–41. [[CrossRef](#)]
33. Yeh, J.-R.; Shieh, J.-S.; Huang, N.E. Complementary Ensemble Empirical Mode Decomposition: A Novel Noise Enhanced Data Analysis Method. *Adv. Adapt. Data Anal.* **2010**, *2*, 135–156. [[CrossRef](#)]
34. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
35. Eberhart, R.; Kennedy, J. A new optimizer using particle swarm theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995; pp. 39–43.
36. Zhang, X.; Liu, H.; Zhang, T.; Wang, Q.; Wang, Y.; Tu, L. Terminal crossover and steering-based particle swarm optimization algorithm with disturbance. *Appl. Soft Comput.* **2019**, *85*, 105841. [[CrossRef](#)]
37. Taylor, K.E. Summarizing multiple aspects of model performance in a single diagram. *J. Geophys. Res. Atmos.* **2001**, *106*, 7183–7192. [[CrossRef](#)]
38. Li, P.; Ma, C.; Ning, J.; Wang, Y.; Zhu, C. Analysis of Prediction Accuracy under the Selection of Optimum Time Granularity in Different Metro Stations. *Sustainability* **2019**, *11*, 5281. [[CrossRef](#)]
39. Dhiman, G.; Kumar, V. Seagull optimization algorithm: Theory and its applications for large-scale industrial engineering problems. *Knowl. -Based Syst.* **2018**, *165*, 169–196. [[CrossRef](#)]
40. Mirjalili, S.; Lewis, A. The Whale Optimization Algorithm. *Adv. Eng. Softw.* **2016**, *95*, 51–67. [[CrossRef](#)]
41. Mirjalili, S.; Mirjalili, S.; Lewis, A. Grey Wolf Optimizer. *Adv. Eng. Softw.* **2014**, *69*, 46–61. [[CrossRef](#)]
42. Dragomiretskiy, K.; Zosso, D. Variational mode decomposition. *IEEE Trans. Signal Processing* **2014**, *62*, 531–544. [[CrossRef](#)]
43. Zhang, Y.; Wen, X.; Jiang, L.; Liu, J.; Yang, J.; Liu, S. Prediction of high-quality reservoirs using the reservoir fluid mobility attribute computed from seismic data. *J. Pet. Sci. Eng.* **2020**, *190*, 107007. [[CrossRef](#)]
44. Bi, J.; Lin, Y.; Dong, Q.; Yuan, H.; Zhou, M. Large-scale water quality prediction with integrated deep neural network. *Inf. Sci.* **2021**, *571*, 191–205. [[CrossRef](#)]
45. Li, Z.; Ge, H.; Cheng, R. Traffic flow prediction based on BILSTM model and data denoising scheme. *Chin. Phys. B* **2022**, *31*, 040502. [[CrossRef](#)]
46. Wang, S.; Zhao, J.; Shao, C.; Dong, C.D.; Yin, C. Truck Traffic Flow Prediction Based on LSTM and GRU Methods With Sampled GPS Data. *IEEE Access* **2020**, *8*, 208158–208169. [[CrossRef](#)]