ISSN 1648-4142 / eISSN 1648-3480

TRANSPORT

2024
Volume 39
Issue 3
Pages 209–228
https://doi.org/10.3846/transport.2024.20538

Original Article

TWICE CLUSTERING BASED HYBRID MODEL FOR SHORT-TERM PASSENGER FLOW FORECASTING

Sheng WANG[1], Xinfeng YANG[2]✉

[1] Gansu Provincial Transportation Development Research Center, Lanzhou, China
[2] School of Traffic and Transportation Engineering, Lanzhou Jiaotong University, Lanzhou, China

Highlights:

- the proposed a novel hybrid model which can effectively improve the accuracy of the forecasting;
- optimizing SVR model parameters through swarm intelligence algorithm;
- presented approach performs TC with respect to values of passenger flow in different time periods;
- using clustering and decomposition algorithms to reduce the volatility of passenger flow data.

Article History:

submitted	13 July 2020;
resubmitted	1 January 2021, 18 March 2021;
accepted	18 May 2021.

Abstract. Short-term metro passenger flow prediction plays a great role in traffic planning and management, and it is an important prerequisite for achieving intelligent transportation. So, a novel hybrid Support Vector Regression (SVR) model based on Twice Clustering (TC) is proposed for short-term metro passenger flow prediction. The training sets and test sets are generated by TC with respect to values of passenger flow in different time periods to improve the prediction accuracy. Furthermore, each obtained cluster is decomposed by using the Complete Ensemble Empirical Mode Decomposition with Adaptive Noise (CEEMDAN) algorithm and the Ensemble Empirical Mode Decomposition (EEMD) algorithm, respectively. The volatility of each component obtained after decomposition is further reduced. Then, the SVR model optimized by the Grey Wolf Optimization (GWO) algorithm is used to predict the decomposed components. Moreover, forecast based on one-month data from Xi'an Metro Line 2 Library Station (China). By comparing the prediction results of the TC condition, the Once Clustering (OC) condition and the non-clustering condition, it shows that the TC approach can adequately model the volatility and effectively improve the prediction accuracy. At the same time, experimental results show that the novel hybrid TC–CEEMDAN–GWO–SVR model has superior performance than Genetic Algorithm (GA) optimized SVR (SVR–GA) model and hybrid Back Propagation Neural Network (BPNN) model.

Keywords: short-term passenger flow forecasting, twice clustering, support vector regression, grey wolf optimization, complete ensemble empirical mode decomposition, adaptive noise.

✉Corresponding author. E-mail: *xinfengyang@mail.lzjtu.cn*

Notations

ANN – artificial neural network;
ARIMA – autoregressive integrated moving average;
BPNN – back propagation neural network;
CEEMDAN – complete EEMD with adaptive noise;
DBI – Davies–Bouldin index;
EEMD – ensemble EMD;
EMD – empirical mode decomposition;
ERM – empirical risk minimization;
GA – genetic algorithm;
GWO – grey wolf optimization;
IMF – intrinsic mode function;
ITS – intelligent transportation systems;
KNN – *k*-nearest neighbour;
KARIMA – kohnon ARIMA;
LS – least-squares;
MAE – mean absolute error;
MAPE – mean absolute percentage error;
OC – once clustering;
RBF – radial basis function;
RMSE – root mean square error;
RNN – recurrent neural network;
SARIMA – seasonal ARIMA;
SRM – structural risk minimization;
STARIMA – space–time ARIMA;

STDI – standard deviation-based index;
SVM – support vector machine;
SVR – support vector regression;
TC – twice clustering.

1. Introduction

In the past few decades, many people in China have selected urban rail transit as their own way of travel for the efficient and convenient of urban rail transit. So, many cities have urban rail transit to take locals and tourists alike rapidly around the city. 520 cities of 75 countries have opened urban rail transit until 2019, with an operating mileage of 28198.09 km. For China, the total mileage of urban rail transit is 6730.27 km. From this point of view, urban rail transit is developing rapidly in the world. Correspondingly, passenger flow of urban rail transit increases year by year. Such as, in 2019, the cumulative number of passengers in Beijing, Shanghai and Guangzhou increased by 1.51%, 4.72% and 9.38%, respectively. At the same time, the average daily passenger flow of Beijing and Shanghai exceeded 10 million passengers in 2019 (10.869 million and 10.643 million passengers, respectively) (CURTA 2019).

Accordingly, excessive passenger demand leads to train congestion and inadequate capacity of urban rail transit facilities (Zhong *et al.* 2016). The successful implementation of traffic forecasting application can achieve proactive passenger flow control, adjust passenger travel routes, modes and time. It can provide effective information for alleviating traffic congestion and improving traffic safety (Ma *et al.* 2015).

Short-term traffic forecasting has become the main part of the research and application of ITS since 1980s. By simulating traffic characteristics such as traffic volume, traffic density and traffic speed, researchers can generate expected traffic conditions and obtain short-term traffic prediction results (Vlahogianni *et al.* 2014). A number of prediction methods have been proposed by researchers. As one of the commonly used model, ARIMA model has been concerned by researchers since the 1970s. The classical ARIMA model is used to predict the road traffic flow (Hamed *et al.* 1995). Considering the periodicity of traffic flow, some studies use the SARIMA model to fit the cyclical characteristics of traffic flow and predict (Kumar, Vanajakshi 2015). In order to improve the prediction effect, some scholars have improved the ARIMA model, such as KARIMA (Van Der Voort *et al.* 1996), subset ARIMA (Lee, Fambro 1999). Based on the variable of historical traffic flow, Williams (2011) used ARIMAX model to predict traffic flow. Kamarianakis & Prastacos (2005) proposed STARIMA model, which takes time and spatial information into account, and integrates them as linear combination weight information into the model, and achieves high robustness. Šliupas (2006) used linear regression and multiple regression to forecasting annual average daily traffic.

The randomness and nonlinearity of traffic flow limit the fitting ability of traditional linear time series model. Machine learning model can effectively fit the historical data, and then predict the future data. In recent years, more and more attention has been paid to the traffic flow prediction model based on machine learning. The KNN regression model has been used by many researchers to construct traffic flow prediction model network. In 1995, the SVM as a new supervised learning model that can realize classification and regression operation was 1st proposed by Cortes & Vapnik (1995). The SVM model performs well in different fields, and its generalization ability is outstanding. SVM model is widely used in traffic forecasting. SVM model preforms well in urban traffic flow forecasting (Li *et al.* 2013), market traffic demand forecasting (Plakandaras *et al.* 2019), road transportation needs (Mrówczyńska *et al.* 2012) ,traffic safety forecasting (Ren, Zhou 2011) and traffic speed forecasting (Wang, Shi 2013). Deng *et al.* (2012) and Li *et al.* (2013) used SVR to predict traffic flow. Hu *et al.* (2011) proposes the LS–SVM model to predict passenger flow. As a non-parametric model, neural networks have excellent properties. In addition, it can not require prior knowledge of the relationship between input sets and output sets when dealing with nonlinear problems (Zhang *et al.* 1998). In recent years, the performance of deep neural network in some fields has greatly exceeded the traditional machine learning methods (Hao *et al.* 2019; Yang *et al.* 2021a). Because of its strong fitting ability, it can automatically learn complex nonlinear features from the original data. Some scholars have tried to use the deep learning model to predict the traffic flow (Yang *et al.* 2017, 2021b; Liu, Chen 2017). Li *et al.* (2019b) proposes a new dynamic RBF neural network to forecast outbound passenger volumes. Although these methods can represent the complex characteristics of traffic flow, they are difficult to fit the sequence dependence of traffic flow. The model based on RNN (Hochreiter, Schmidhuber 1997) has solved the above problems.

It is worth noting that many scholars are committed to the study of traffic flow prediction technology of hybrid model, hoping to combine the characteristics of different models to play a more powerful advantage. Zhang *et al.* (2018) composes traffic flow series into 4 different components, 4 parts are treated and modeled separately by using different methods, such as spectral analysis, time series and statistical volatility analysis. Zhang (2003) combines ARIMA model with neural network. Firstly, ARIMA model is used to analyze the linear part of the sequence, and then neural network is used to model the residual part. GWO algorithm (Mirjalili *et al.* 2014) is a swarm intelligence algorithm proposed in 2014, which has achieved good performance in air pollution forecasting (Zhu *et al.* 2019), tumor evaluation (Ramakrishnan, Sankaragomathi 2017), carbon emission forecasting (Heydari *et al.* 2019) and so on.

In order to achieve more accurate prediction, support the urban rail transit operation and achieve proactive control of the passenger flow, this article attempts to develop a short-term passenger flow prediction method that combines the GWO algorithm and SVM regression under the condition of TC and decomposition.

The main innovation of this article is to propose a novel hybrid SVR–GWO model based on TC to predict urban rail transit passenger flow. Due to the time-varying characteristics of passenger travel demand, the original passenger flow data show different characteristics in different time periods. Using the TC method can effectively adequately model the volatility and ensure the accuracy of subsequent predictions. At the same time, the SVR model is good at processing nonlinear and non-stationary data, but the relevant parameters of the model are difficult to determine. Combining the GWO algorithm with the SVR model can quickly and effectively determine the parameters of the prediction model and reduce the prediction error. Through experiments, we found that the hybrid model mentioned in this article performs well and can accurately predict short-term urban rail transit passenger flow.

The rest of this article is as follows:

- the introduction is presented in current Section 1;
- the passenger flow analysis and model framework is drawn in Section 2;
- the methodology is reviewed in Section 3;
- the empirical study is presented in Section 4;
- the conclusion is drawn in Section 5.

2. Passenger flow analysis and model framework

The change of passenger flow is a reflection of urban social and economic activities. The core of passenger flow analysis is to analyze the temporal and spatial distribution of passenger flow.

In terms of passenger flow time distribution, hourly passenger flow is used to determine the number of urban rail transit entrances and exits, passages and other equipment, which is the basis for calculating the full-day driving plan and vehicle deployment plan. The hourly passenger flow changes with the rhythm of urban life. At night, there is very little passenger flow, and it gradually increases at dawn. The working hours reach a peak, then the passenger flow gradually decreases, and there is a 2nd peak when off work. The passenger flow gradually decreases when entering the evening. According to different types of passenger flow distribution, common types are one-way peak type, 2-way peaks type, full peak type, hump type and no peak type. Among them, in the one-way peak type, the distribution time of passenger flow is relatively concentrated. In the 2-way peaks type, there are 2 peaks corresponding to the morning peak and the evening peak. In the full peak type, the passenger flow is relatively large throughout the day. In the hump type, passenger flow fluctuates with the start and end of large-scale events. In the no peak type, passenger traffic is small throughout the day. Take the passenger flow of Xi’an Metro Line 2 Library Station as an example (as shown in Figure 1). Due to the characteristics of commuters’ demand, the distribution of passenger flow at working days is obviously 2-way peaks type. The passenger flow distribution corresponds to 2 peaks in the morning and in the evening. The passenger flow distribution is full peak type at the weekend. There is no obvious trough in passenger flow distribution, and no morning and evening peak. Meanwhile, Because of a large portion of urban railway transit passengers are commuters, the passenger flow on working days is significantly higher than that at weekend. Since people’s work and rest are carried out on a weekly cycle, the regularity of this activity must be reflected in the changes in passenger flow on each day of the week. On the rail transit line with commuting passenger flow, the passenger flow on weekends will be reduced. On the rail transit lines connecting commercial outlets and tourist attractions, the passenger flow on weekends tends to increase. According to the unbalanced and regular changes in the distribution of the full-day passenger flow in a week, from the perspective of operational economy, the rail transit system often implements different full-day train schedules and train schedules within a week. Figure 2 shows the peak hour passenger flows of a certain station in Xi’an urban rail transit in March.

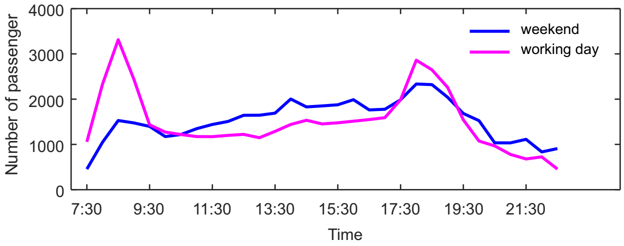

Figure 1. Fluctuation of passenger flow on working day and weekend

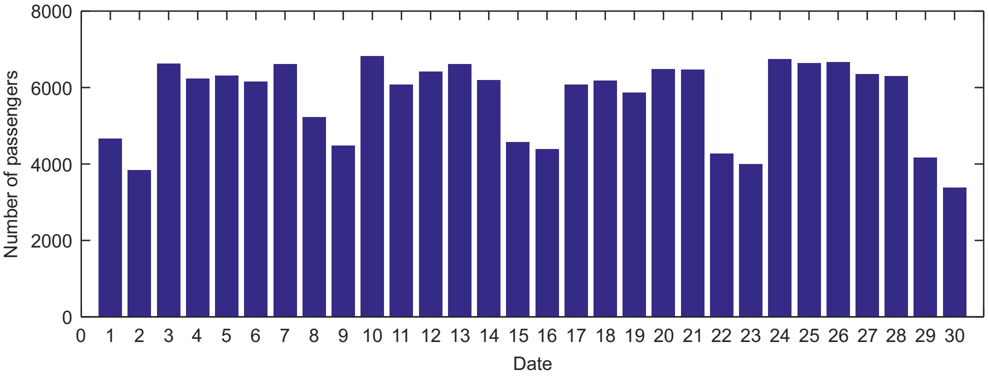

Figure 2. Peak hour passengers in Xi’an Metro Line 2 Library Station

In terms of the spatial distribution of passenger flow, each line of the urban rail transit network has different passenger flow scales and distribution laws due to the different urban passenger flow corridors where it is located and the nature of the land along the line. On rail transit lines, due to the direction of passenger flow, the passenger flow in the upward and downward directions is usually not equal. On radial rail transit lines, the imbalance of passenger flow in the upward and downward directions in the morning and evening peak hours is particularly obvious. On rail transit lines, due to the different scales and numbers of passenger flow distribution points covered by the route's travel area, the number of passengers on and off at each station of the line is different. It is inevitable that there is an imbalance in the passenger flow of each section. It is not uncommon for the number of passengers on and off at various stations of rail transit lines to be uneven. On many lines, the total amount of boarding and landing at each station of the whole line is concentrated in a few stations. In addition, the scale of new residential areas and the commissioning of new rail transit lines will also cause major changes in station rides and landings.

In view of the above-mentioned characteristics of passenger flow, this article considers the use of clustering to adequately model the volatility at different time periods. Cluster passenger flow with similar characteristics to improve the accuracy of passenger flow prediction. The model framework of this article is shown in the Figure 3:

- *Step 1*: collect raw passenger flow data;
- *Step 2*: perform the OC of passenger flow data according to the size of the daily peak traffic;
- *Step 3*: the data after the OC is then clustered for the 2nd time according to the characteristics of passenger demand in different time periods;
- *Step 4*: use the CEEMDAN method and EEMD method to decompose different clusters;
- *Step 5*: hybrid GWO–SVR model is used to predict the decomposed components;
- *Step 6*: assemble the forecast results to obtain the final forecasting value.

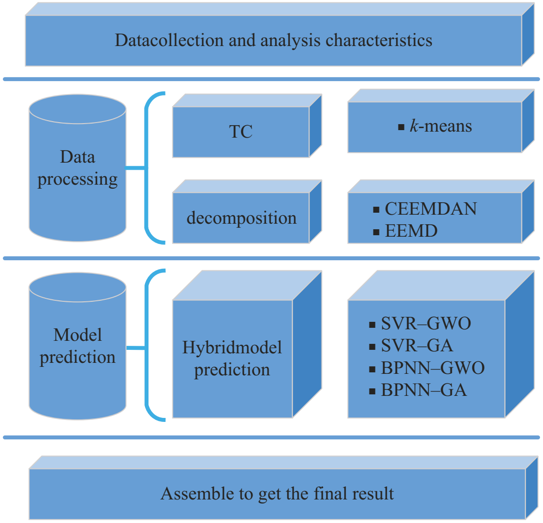

Figure 3. Model framework

3. Materials and methods

3.1. *k*-means algorithm

Cluster analysis attempts to divide the sample set into a series of disjoint subsets. Each subset is called a "cluster" (Wang *et al.* 2018a). The *k*-means algorithm is a kind of prototype-based clustering and is known for its high efficiency. The *k*-means algorithm calculates the category center C_i of each cluster by dividing the data into *k* cluster. The objective function is expressed as follows:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} dist\left(p, C_i\right)^2, \tag{1}$$

where: *p* is the point belonging to the specified cluster *i*; C_i is the centroid of the cluster *i*.

In the article, the initial mean vector is *k* samples randomly selected from the sample set. The optimal number of clusters *k* is determined by comparing the DBI and the STDI. The smaller the DBI value and the larger the STDI value, the better the clustering effect:

$$DBI = \frac{1}{k} \cdot \sum_{p=1}^{k} \max\left(\frac{S_p + S_q}{\left|C_p - C_q\right|}\right), \tag{2}$$

where: S_p, S_q represent the standard deviation of the Euclidean distance between the elements in Cluster *p* and Cluster *q*, respectively;

$$STDI = \frac{\frac{1}{k} \cdot \left(\sum_{p=1}^{k} \left\|C_p - \bar{x}\right\|^2\right)}{\sum_{p=1}^{k} \frac{1}{n_p} \cdot \left(\sum_{u=1}^{n_p} \left\|x_u - C_p\right\|^2\right)}, \tag{3}$$

where: C_p denotes the centroid of Cluster *p*; $\bar{x}$ is the centroid of all samples; x_u is the *u*th sample of Cluster *p*; n_p denotes the number of samples of Cluster *p*.

3.2. EEMD

Originally, the EMD algorithm was a data decomposition method proposed by Huang *et al.* (1998), which can adaptively decompose the original sequence. The EMD algorithm can well handle nonlinear and non-stationary time series while retaining the original sequence characteristics. However, in the process of signal decomposition, modal aliasing will occur (Li *et al.* 2019a).

Wu & Huang (2009) proposed the EEMD algorithm to solving shortcoming of the traditional EMD algorithm. The EEMD algorithm projects sequence components of different scales onto appropriate reference scales by adding uniformly distributed white noise (Lei *et al.* 2013). EEMD algorithm avoids mode aliasing and retains the advantage that EMD method can decompose time series directly without any basis function and filter function (Wu *et al.* 2019). It is an effective method to cope with time series.

The EEMD algorithm obtains a series of IMF and a residual during the adaptive decomposition of the original time series.

The EEMD algorithm is developed based on the EMD algorithm. The EMD algorithm is described as follows:

- *Step 1*: the local maximum points of the time series $x(t)$ constitute the upper envelope, and the local minimum points constitute the lower envelope;
- *Step 2*: find the average value of the upper and lower envelopes as $mean(t)$, and calculate $h(t)$:

$$mean(t) = \frac{z_u v(t) + z_l(t)}{2}; \tag{4}$$

$$h(t) = y(t) - mean(t); \tag{5}$$

- *Step 3*: repeat k times until $h(t)$ satisfies the conditions of the IMF component and treats $h(t)$ as the 1st IMF component c_1;
- *Step 4*: calculate the residue r_1:

$$r_1 = x(t) - c_1; \tag{6}$$

- *Step 5*: set the residue r_1 to the new time series $x(t)$ and repeat Steps 1…5 to get all the IMF components c_j and one residue r_n after decomposition.

The final original time series can be expressed as:

$$x(t) = \sum_{j=1}^{n} c_j + r_n . \tag{7}$$

The EEMD algorithm is described as follows:

- add different Gaussian white noise to the original signal, so that a new number of i signals is reconstructed after adding the noise:

$$x_i(t) = x(t) + w_i(t), \tag{8}$$

where: $x(t)$ is the original signal; $w_i(t)$ is the ith Gaussian white noise, i = 1, ..., I; $x_i(t)$ is a new signal formed by adding noise;

- decompose $x(t)$ by the EMD algorithm to obtain a series of modal functions c_{ij} and a unique residual term $r_i(t)$;
- average the modal functions in the same order obtained by decomposing the original signal with different Gaussian white noise as the final IMF:

$$c_j = \frac{1}{I} \cdot \sum_{i=1}^{i} c_{ij} . \tag{9}$$

3.3. CEEMDAN

The CEEMDAN algorithm has been improved on the basis of the EEMD algorithm. The process is as follows:

- decompose the original signal using the EEMD algorithm to obtain the 1st modal component:

$$\bar{c}_1 = \frac{1}{I} \cdot \sum_{i=1}^{I} c_{ik} ; \tag{10}$$

- calculate the remainder of the 1st phase, where k = 1:

$$r_1(t) = x(t) - \bar{c}_1 ; \tag{11}$$

- construct a signal $r_k(t) + \varepsilon_k \cdot E_k(w_i(t))$, where $r_k(t)$ represents the remainder of the signal at stage; ε_k is the signal-to-noise ratio; $E_k(\cdot)$ represents the kth modal component of the signal after EMD decomposition. The following formula represents the (k + 1)th modal k component obtained by decomposing the structural signal into the 1st modal component:

$$\bar{c}_{k+1} = \frac{1}{I} \cdot \sum_{i=1}^{I} E_1\left(r_k(t) + \varepsilon_k \cdot E_k\left(w_i(t)\right)\right), \quad k = 1, \ldots, K ; \tag{12}$$

- calculate the remainder of the signal $r_{k+1}(t)$:

$$r_{k+1}(t) = r_k(t) - \bar{c}_{k+1} ; \tag{13}$$

- let k increase by 1 and repeat Steps 3…5 until the remaining terms are not suitable for further decomposition:

$$\bar{R}(t) = x(t) - \sum_{k=1}^{K} \bar{c}_k . \tag{14}$$

3.4. SVR model

The SVM model was originally used for classification purposes, and it is a good performance in regression by extending (Hong 2011). The ERM principle is used in the general artificial intelligence model. But ERM principle does not ensure accurate forecasting on the test set. SVR minimizes the boundaries of generalization errors through SRM rather than making the mean square error on the training dataset minimized. The SVR model maps the training set data $\{(x_i, y_i)\}$ through the nonlinear map y, so that y is related to $f(x)$ and ε, where ε represents random noise:

$$f(\mathbf{x}) = \boldsymbol{\omega}^{\mathbf{T}} \cdot \varphi(\mathbf{x}) + b, \tag{15}$$

where: $\boldsymbol{\omega}\left(\boldsymbol{\omega} \in \Re_h^n\right)$ and $b(b \in \Re)$ are parameters; $f(\mathbf{x})$ represents the predicted result value.

For empirical risk $R_{emp}(f)$ is considered to be the average of the loss function values for each point in the training set:

$$R_{emp}(f) = \frac{1}{n} \cdot \sum_{i=1}^{n} l\left(y_i, f\left(\mathbf{x}_i, \mathbf{w}\right)\right), \tag{16}$$

where: $l(y, f)$ represents the loss function.

In this article, the ε (insensitive loss function) is recorded as $\phi_\varepsilon(\mathbf{y}, f(\mathbf{x}))$, which is defined as follows:

$$\phi_\varepsilon(\mathbf{y}, f(\mathbf{x})) = \begin{cases} \left|f(\mathbf{x}) - \mathbf{y}\right| - \varepsilon, & \text{if } \left|f(\mathbf{x}) - \mathbf{y}\right| \geq \varepsilon; \\ 0, & \text{otherwise.} \end{cases} \tag{17}$$

According to the above formula, the empirical risk function is expressed as:

$$R_{emp}(f) = \frac{1}{n} \cdot \sum_{i=1}^{n} \phi_\varepsilon\left(y_i, f\left(\mathbf{x}_i\right)\right). \tag{18}$$

The ε (insensitive loss function) maximizes the distance between the 2 subsets that are separated from the optimal hyperplane found in the high-dimensional feature space.

The regression problem is finally expressed as a quadratic minimization problem:

$$\min_{\mathbf{w},\, b,\, \xi^*,\, \xi} R_{\varepsilon}\left(\mathbf{w}, \xi^*, \xi\right) = \frac{1}{2}\cdot \mathbf{w}^{\mathbf{T}} \cdot \mathbf{w} + C \cdot \sum_{i=1}^{n}\left(\xi_i^* + \xi_i\right) \tag{19}$$

subject to:

$$-\mathbf{w}^{\mathbf{T}} \cdot \varphi\left(\mathbf{x}_i\right) - b \le \varepsilon + \xi_i^*;$$

$$-\mathbf{y}_i + \mathbf{w}^{\mathbf{T}} \cdot \varphi\left(\mathbf{x}_i\right) + b \le \varepsilon + \xi_i;$$

$$\xi_i \ge 0;\ \ \xi_i^* \ge 0;\ \ i = 1, 2, \ldots, n, \tag{20}$$

where: $\xi = \left(\xi_1, \ldots, \xi_n\right)$, $\xi^* = \left(\xi_1^*, \ldots, \xi_n^*\right)$ represent the slack variable of the upper and lower constraints; C is a constant "cost" parameter greater than 0, which is used to specifies the tradeoff between the risk function minimization and smoothness (Schölkopf, Smola 2001).

The SVR model dual optimization problem is represented by the Lagrangian function as follow:

$$\max_{\alpha, \alpha^*}\left(-\frac{1}{2}\cdot \sum_{i=1}^{n}\sum_{j=1}^{n}\left(\beta_i - \beta_i^*\right)\cdot\left(\beta_j - \beta_j^*\right)\cdot K\left(\mathbf{x}_i, \mathbf{x}_j\right) - \varepsilon \cdot \sum_{i=1}^{n}\left(\beta_i + \beta_i^*\right) + \sum_{i=1}^{n} y_i \cdot \left(\beta_i - \beta_i^*\right)\right) \tag{21}$$

subject to:

$$\sum_{i=1}^{n}\left(\beta_i - \beta_i^*\right) = 0\,,$$

$$0 \le \beta_i \le C,\ \ 0 \le \beta_i^* \le C,\ \ i = 1, \ldots, n\,, \tag{22}$$

where: $\beta = \left(\beta_1, \ldots, \beta_n\right)$, $\beta^* = \left(\beta_1^*, \ldots, \beta_n^*\right)$ represent the positive and negative Lagrangian multipliers, respectively; the value of the kernel function $K\left(\mathbf{x}_i, \mathbf{x}_j\right)$ is the inner product of the 2 vectors $\left(\mathbf{x}_i, \mathbf{x}_j\right)$ in the respective feature spaces $\varphi\left(\mathbf{x}_i\right)$, $\varphi\left(\mathbf{x}_j\right)$.

At present, there is not any uniform standard for the selection of kernel function. The Gaussian kernel function is a function that satisfies the Mercer's condition. Because of its simplicity, the data can be nonlinearly mapped to the finite dimensional feature space. This kernel function is employed in the article.

The Gaussian kernel function is expressed as:

$$K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \exp\left(-0.5 \cdot \frac{\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2}{\sigma^2}\right). \tag{23}$$

3.5. BPNN

ANN was proposed in the 1940s, which is a connection model that simulates the human brain. Rumelhart and Hinton (Rumelhart *et al.* 1986) discovered BPNN in 1986. Because of excellent performance, BPNN has gradually become a model that has been most studied by researchers. BPNN has strong non-linear mapping capabilities. It has been proven that a 3-layer BPNN can approximate a continuous function with arbitrary precision (Cybenko 1989). Figure 4 shows the topology of the neuron model. $\mathbf{X} = \left\{x_1, \ldots, x_n\right\}$ represents the information of the data, $\mathbf{W} = \left\{w_{i1}, \ldots, w_{in}\right\}$ represents the weight of each input, x_0 represents the threshold, and $F(x)$ is the excitation function.

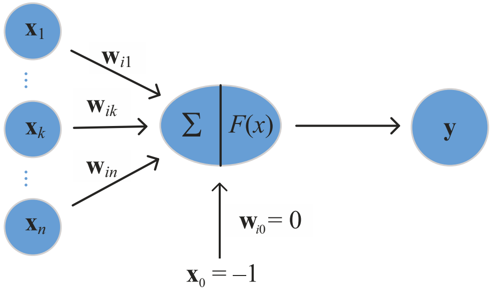

Figure 4. Neuron model topology

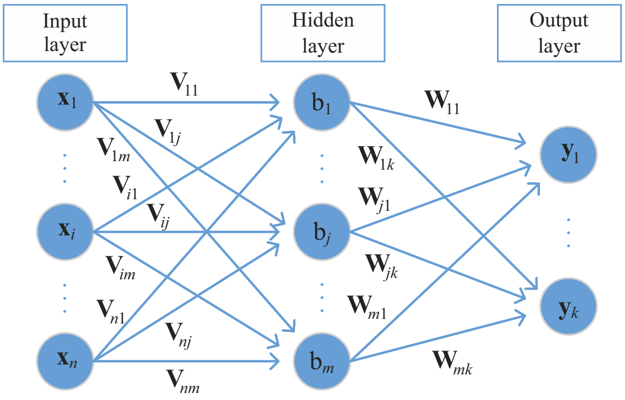

Figure 5. BPNN network structure diagram

BPNN implements data processing through forward propagation of the input data and back propagation of the error. Taking a 3-layer BPNN as an example (showed in Figure 5), during the forward propagation of input data, the information is input through the input layer. Then the information is processed through the weights and thresholds of the hidden layer. And finally the information is output at the output layer. In the process of backpropagation, for the output value that does not meet the expectations, the error signal is backpropagated. The connection weights and thresholds are modified layer by layer, so that the new round of output results meet the expectations.

3.6. GWO

GWO algorithm is a new swarm intelligence algorithm proposed by simulating wolves hunting in 2014. Compared with the excellent existing algorithms, GWO algorithm has been proved to perform better (Sulaiman *et al.* 2015). As one of the latest bio-inspired optimization algorithms, GWO algorithm simulates the behaviour of wolves in the wild hunting, which is an algorithm with social hierarchical nature. The wolves are divided according to the social dominant hierarchy:

- α *wolf*: as the top leader of the wolves, it is responsible for leading the wolves for hunting and is responsible for the main decisions of the wolves. The α wolf is not necessarily the strongest wolf in the wolves, but in terms of management ability, the α wolf must be the best;

- β *wolf*: a 2nd-level wolf who is responsible for transmitting information about the α wolf and supporting the wolves, and becomes a new generation of leader candidates after the death or aging of the α wolf;
- δ *wolf*: the 3rd-level wolf, obeying the α wolf and the β wolf, is responsible for the unified coordination within the wolves and weakens the inner divisions of the wolves;
- ω *wolf*: the 4th-level of wolf, which usually needs to obey the wolves of other social hierarchy.

The GWO algorithm is constructed according to the behaviour of the wolves. Wolves approach their prey (optimal solution) in search space by 3 individual wolf (α, β and δ). Through multiple iterations, the optimal solution keeps getting closer to the global optimal solution. GWO algorithm is described as follow (Mirjalili *et al.* 2014):

- *social hierarchy*: in the GWO algorithm, the pros and cons of each scheme are identified according to the wolf group social hierarchy. Correspondingly, α is the best solution, β and δ are suboptimal solutions, and the remaining solutions are ω;
- *encircling the prey*: the following equations describe the behaviour encircling the prey:

$$\mathbf{D} = \left|\mathbf{C} \cdot \mathbf{X}_p\left(t\right) - \mathbf{X}\left(t\right)\right|; \tag{24}$$

$$\mathbf{X}\left(t+1\right) = \mathbf{X}_p\left(t\right) - \boldsymbol{A} \cdot \mathbf{D}, \tag{25}$$

where: **D** represents the distance of the current wolf from the prey; *t* represents the number of iterations; **X** is the coordination of the wolf; **A**, **C** represent the coefficient vector; $\mathbf{X}_p$ represents the coordination of prey.

The expressions of coefficient vectors as follows:

$$\mathbf{A} = 2 \cdot \mathbf{a} \cdot \boldsymbol{r}_1 - \mathbf{a}; \tag{26}$$

$$\mathbf{C} = 2 \cdot \boldsymbol{r}_2, \tag{27}$$

where: **a** decreases linearly from 2 to 0; $\boldsymbol{r}_1$, $\boldsymbol{r}_2$ is a random vector within [0, 1];

- *hunting*: assuming that the head wolf knows more information about the potential location of the prey (the optimal solution), and retain the position of the previous α wolf, β wolf and δ wolf, i.e. the closest location to the prey. According to social hierarchy of the wolf, the remaining wolves ω are updated and adjusted by α wolf, β wolf and δ wolf. The process is represented as following:

$$\mathbf{D}_\alpha = \left|\boldsymbol{C}_\alpha \cdot \mathbf{X}_\alpha\left(t\right) - \mathbf{X}\left(t\right)\right|; \tag{28}$$

$$\mathbf{D}_\beta = \left|\boldsymbol{C}_\beta \cdot \mathbf{X}_\beta\left(t\right) - \mathbf{X}\left(t\right)\right|; \qquad (29$$

$$\mathbf{D}_\delta = \left|\boldsymbol{C}_\delta \cdot \mathbf{X}_\delta\left(t\right) - \mathbf{X}\left(t\right)\right|; \tag{30}$$

$$\mathbf{X}_1 = \mathbf{X}_\alpha - \boldsymbol{A}_1 \cdot \mathbf{D}_\alpha; \tag{31}$$

$$\mathbf{X}_2 = \mathbf{X}_\beta - \boldsymbol{A}_2 \cdot \mathbf{D}_\beta; \tag{32}$$

$$\mathbf{X}_3 = \mathbf{X}_\delta - \boldsymbol{A}_3 \cdot \mathbf{D}_\delta; \tag{33}$$

$$\mathbf{X}_p\left(t+1\right) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3}, \tag{34}$$

where: $\mathbf{X}_\alpha$, $\mathbf{X}_\beta$, $\mathbf{X}_\delta$ represent the current position vector of α wolf, β wolf and δ wolf; $\mathbf{D}_\alpha$, $\mathbf{D}_\beta$, $\mathbf{D}_\delta$ represent the calculated distance between the wolf and the 3 headed wolves; $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ are random coefficient vectors; *t* represents the current number of iterations;

- *attacking the prey (exploitation)*: when everything was prepared, the wolf began to hunt. The mathematical form of this process is expressed as the value of **a** decreasing linearly from 2 to 0. At the same time, the value of **A** also decreases. The wolf performs the attack behaviour when the value of |**A**| is in the interval [–1, 1];
- *search for prey (exploration)*: the wolves separate from each other while searching for prey, and encircle the α, β and δ when attacking the prey. When |**A**| > 1, the wolves are far away from the prey, which ensures that GWO algorithm has global search ability. While at |**A**| > 1, the wolf approaches the prey. The GWO algorithm iterates continuously until the termination condition is satisfied.

3.7. The hybrid SVR model

This article proposes a hybrid SVR model to predict passenger flow data after processed. Combining GWO algorithm with SVR model to optimize learning parameters of the SVR model. The hybrid model fitness function is expressed as following:

$$fitness = \frac{1}{N} \cdot \sum_{i=1}^{N} \left(\hat{y}_i - y_i\right)^2, \tag{35}$$

where: *N* represents the number of training sets; $\hat{y}_i$ represents the training set prediction value; y_i represents the training set actual value.

Table 1 shows pseudo-code of GWO optimized SVR model.

Table1. Pseudo-code of GWO optimized SVR model

```
Algorithm – GWO optimized SVR model
Initialize wolf $X_i$ ($i$ = 1, 2, ..., $n$)
Initialize the maximum number of iterations $T$
Initialize parameters a, A and C
Set the SVR model parameters according to $X_i$, and make the prediction error of the SVR model as the search agent fitness
$X_1$ – the best search agent
$X_2$ – the 2nd best search agent
$X_3$ – the 3rd best search agent
While $t < T$ do
For each search agent do
    update the position of current search agent by Equation (34)
    calculate the fitness of current search agent after updating the position
End for
Update a, A and C by Equations (26) and (27)
Update $X_1$, $X_2$, $X_3$
$t = t + 1$
End while
Return optimal solution
```

4. Experimental study

This article collects outbound passenger flow data of Xi'an Metro Line 2 Library Station as the original data. It is collected 1–30 March of a certain year at intervals of 30 min. The operating period of urban rail transit is generally consistent with the travel period of urban residents, showing the characteristics of large passenger flow during the day and zero passenger flow at night. Therefore, this article excludes the time period when the passenger flow is zero, and selects the passenger flow data from 7:00 to 22:30 as the original data. In order to speed up the learning speed and convergence speed when training the model, we performed a min-max normalization operation to scale the data to the range of [0, 1].

The original data added up to 930 observations. In the training phase, we select 80% of the original data as the training set (744 observations), and the remaining 20% of the data is used to verify the performance of the model (186 observations). In various literature, error criteria are used to determine whether the model is effective. At present, there is a lack of uniform standards to using error criteria methods, so researchers mostly use different kinds of evaluation indicators to evaluate the effectiveness of models (Niu *et al.* 2016).The hybrid model proposed is evaluated by 3 main evaluation criteria, which are MAPE, RMSE and the MAE. These 3 evaluation criteria are showed as:

$$MAPE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| \frac{y_i - \hat{y}_i}{y_i} \right| \cdot 100\%; \tag{36}$$

$$SE = \sqrt{\frac{\sum_{i=1}^{n} \left(y_i - \hat{y}_i \right)^2}{n}}; \tag{37}$$

$$MAE = \frac{1}{n} \cdot \sum_{i=1}^{n} \left| y_i - \hat{y}_i \right|, \tag{38}$$

where: n is the number of test samples; y_i. is the actual value; $\hat{y}_i$ is the predicted value.

At the same time, all experiments are performed on a personal computer with 2.60GHz for CPU and 16.0GB for RAM.

4.1. Data clustering and data decomposition

In the proposed a novel hybrid model, the 1st step is to divide the data according to the daily passenger flow characteristics. According to peak hour passenger flow on each day mentioned above, passenger flow is clustered by the *k*-means method. The key to the *k*-means algorithm is to determine the DBI value and STDI value. The DBI value (0.5961) and STDI value (4.1594) obtained during the OC process are shown in Table 2. It can be found that when the number of clusters K = 2, the value of DBI is the smallest and the value of STDI is the largest, so the number of clusters K = 2 is the most suitable. Therefore, the passenger flow in March realized the OC, and the clustering results are shown in Table 3. After the OC, the original data is divided into 2 clusters, where Cluster 1 includes passenger flow on working days (Monday, Tuesday, Wednesday, Thursday and Friday), and Cluster 2 includes passenger flow data on weekends (Saturday and Sunday) (showed in Figure 6). This is consistent with the results of passenger flow analysis.

Secondly, after the OC, passenger data sets of Cluster 1 and Cluster 2 are formed. As can be seen from the above, the number of passenger flow is different at different periods on each day. In order to adequately model the volatility and improve the accuracy of the prediction, performing the TC on Cluster 1 and Cluster 2 by *k*-means. The *k*-means method is used to perform clustering for Cluster 1. The DBI and STDI values are calculated when K = 2, 3, 4, 5. Selecting the value of K with the smallest DBI value and the largest STDI value as the number of the TC (as shown in Table 4). When K = 3, the DBI value for Cluster 1 is the smallest (0.6512) and the STDI value is largest (3.5852). Therefore, K = 3 is selected as the number of the TC for Cluster 1. After the TC, Cluster 1 divided into 3 cluster (Cluster 11, Cluster 12 and Cluster 13) (as shown in Figure 7), and corresponding periods are shown in Table 5. At the same time, do the same for Cluster 2. The K value with the smallest DBI value (0.7788) and the largest STDI value (9.1913) is calculated (as shown in Table 6). It can be found that the DBI value of the 2nd clustering is equal to 0.7788 when K = 5, which is the smallest of all DBI values. And the STDI value obtained at same condition is 9.1913, which is the largest compared with other values. Selecting K = 5 as the number of the TC of Cluster 2. Cluster 2 generates 5 new clusters after the TC, which are Cluster 21, Cluster 22, Cluster 23, Cluster 24 and Cluster 25 (as shown in Figure 8). The time periods corresponding to each sub-clusters are shown in Table 7.

Table 2. The number of clusters K of the OC and the corresponding DBI and STDI

K	2	3	4	5
DBI	0.5961	2.4199	4.8838	1.7243
STDI	4.1594	3.845	3.5299	2.9992

Table 3. Results of OC of each cluster

Name	Result
Cluster 1	Monday, Tuesday, Wednesday, Thursday, Friday
Cluster 2	Saturday, Sunday

Table 4. The value of the TC number K of Cluster 1 and corresponding DBI and STDI

K	2	3	4	5
DBI	0.7592	0.6512	1.3301	2.4032
STDI	2.6211	3.5852	3.0733	2.5443

Table 5. Time period and number of the TC of Cluster 1

Name of twice cluster	Time period	Number
Cluster 11	7:00–7:30, 20:00–22:30	120
Cluster 12	9:00–17:30, 19:00–20:00	380
Cluster 13	7:30–9:00, 17:30–19:00	120

Table 6. The value of the TC number K of Cluster 2 and corresponding DBI and STDI

K	2	3	4	5	6
DBI	0.8245	0.9167	0.8889	0.7788	1.2084
STDI	1.0963	1.923	3.3444	9.1913	8.4562

Table 7. Time period of the TC of Cluster 2

Name of twice cluster	Time period	Number
Cluster 21	7:00–7:30	10
Cluster 22	7:30–8:00, 20:30–22:30	50
Cluster 23	8:00–12:30, 19:30–20:30	110
Cluster 24	12:30–17:30, 19:00–19:30	110
Cluster 25	17:30–19:00	30

Thirdly, after the above TC of the original data, a series of clusters are obtained. The clusters are decomposed using the EEMD algorithm and the CEEMDAN algorithm. Each of clusters is decomposed to a series of IMF components $\left(IMF = \left\{imf_1, imf_2, ..., imf_n\right\}\right)$, which is containing the original time series features. These components are distributed from high frequency to low frequency. Figure 9 shows the decomposition results of Cluster 11 using EEMD algorithm and CEEMDAN algorithm.

4.2. Forecasting and models comparison

Comparing with other different models, the effectiveness of the hybrid SVR–GWO model under the above TC and decomposition conditions is illustrated.

The EEMD algorithm or CEEMDAN algorithm is performed on the data under TC. Decomposed IMF components are predicted by the hybrid SVR–GWO model and the hybrid BPNN–GWO model using multi-step method. Among them, the prediction model combines the GWO algorithm with the SVR model (SVR–GWO) and with the

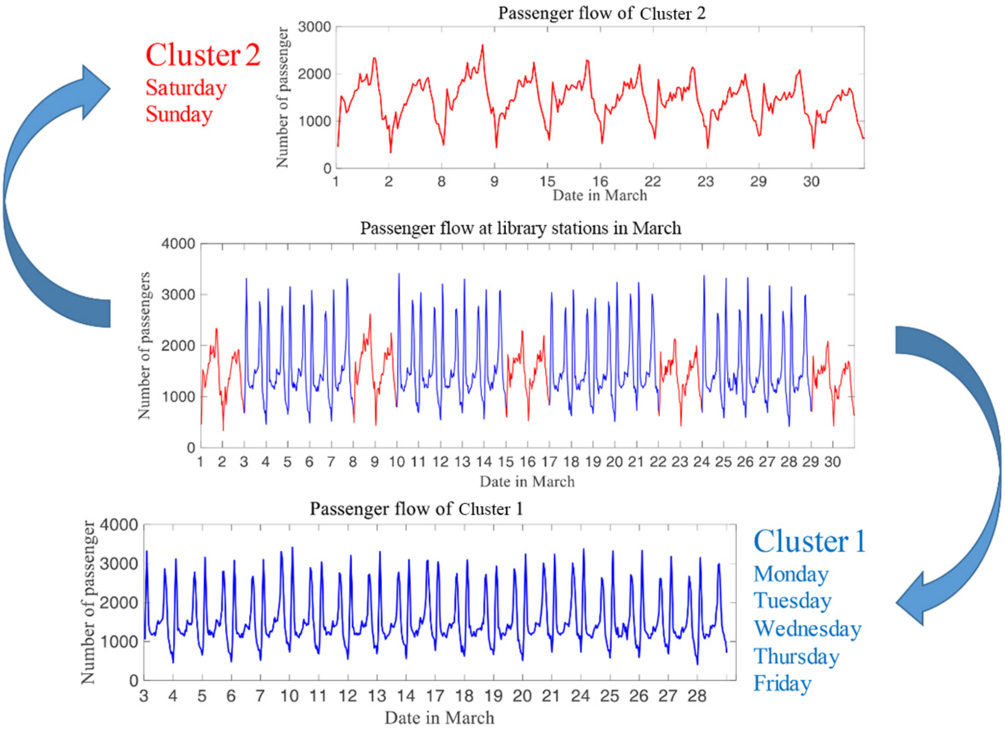

Figure 6. OC result of passenger flow at library stations in March of a certain year

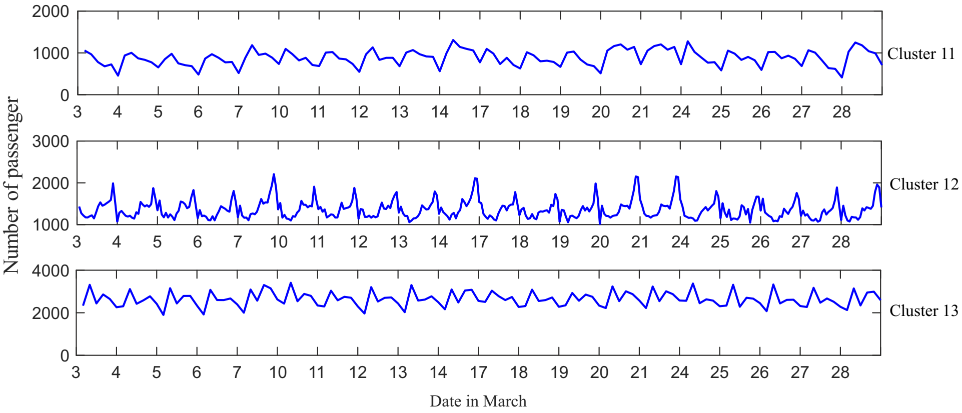

Figure 7. The TC result graph of Cluster 1

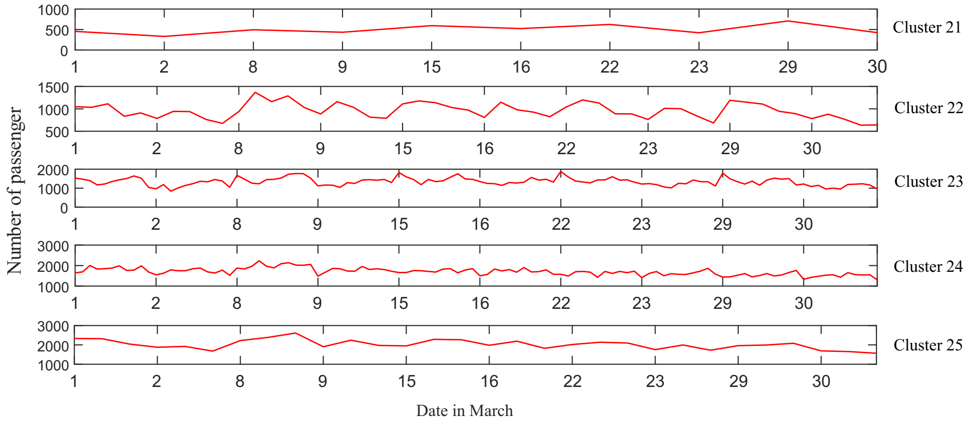

Figure 8. The TC result graph of Cluster 2

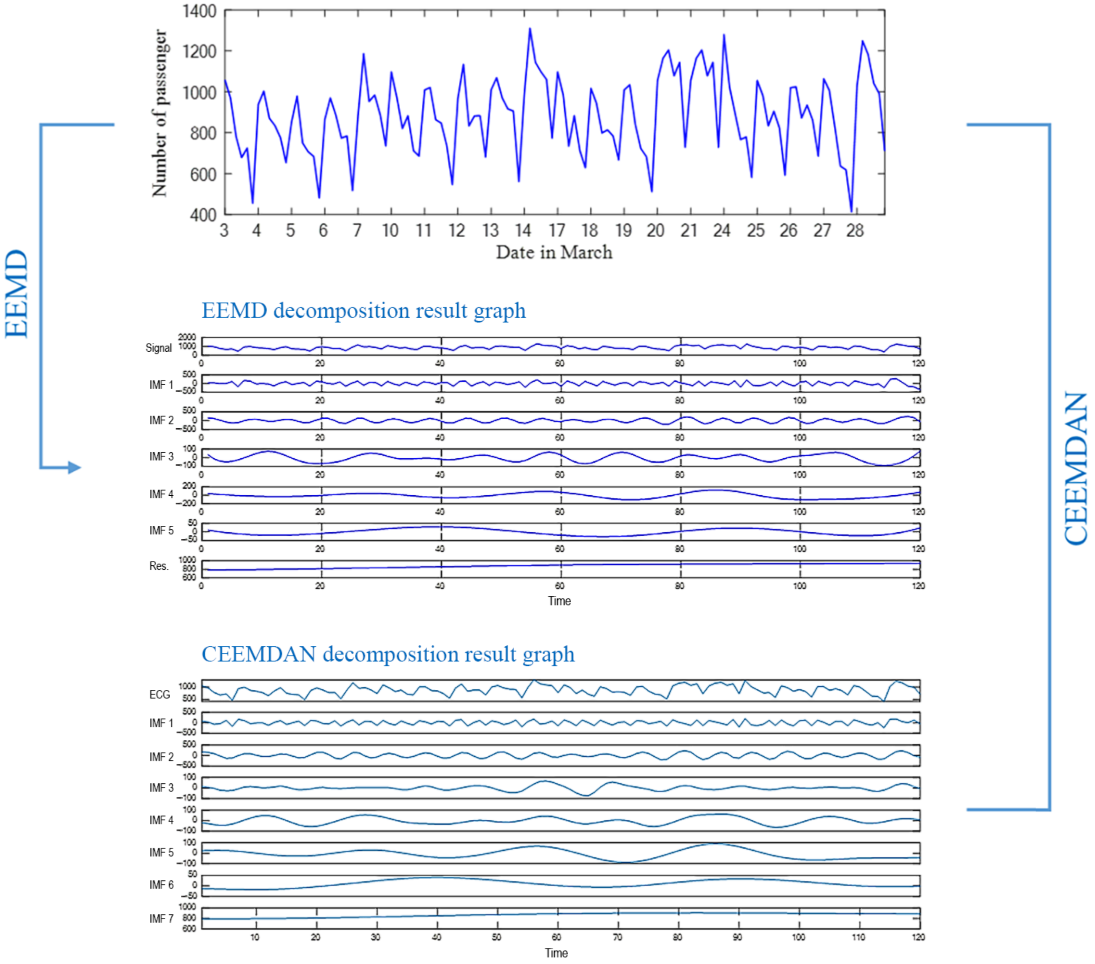

Figure 9. EEMD decomposition and CEEMDAN decomposition of Cluster 11

BPNN model (BPNN–GWO), respectively. The accuracy of the SVR model prediction is mainly affected by the kernel parameters c and σ. How to find the optimal parameters (c, σ) in the solution space is the key to achieving accurate predictions. GWO algorithm can quickly and effectively find the optimal parameter combination of the SVR model in the entire solution space, and avoid falling into a local optimal situation. The parameters of the GWO algorithm are set as follows: the number of wolves n = 15, the number of iterations T = 30, the value of parameter c ranges from (0, 200], the value of parameter σ ranges from (0, 200]. Initial parameters are randomly generated. For BPNN model, the initial value of the weight and threshold of the connections between the neurons in each layer has a direct impact on the final prediction accuracy of the network. Generally, the initial values of the weights and thresholds are randomly generated, which will greatly increase the training time and the number of cycles of the network. Therefore, combining

GWO algorithm with BPNN model, it can effectively shorten model training time and improve prediction accuracy. The parameters setting of GWO algorithm are the same as above. At the same time, the prediction results of GA optimized SVR model (SVR–GA) and GA optimized BPNN model (BPNN–GA) are calculated under the same situation. The parameters of the GA algorithm are set as follows: the maximum number of evolution iterations *maxga* = 200, the maximum number of population size is 20, the cross probability is 0.3, and the mutation probability is 0.1. Then, the influence of different clustering conditions on the predicted results is analysed:

- *different step size*: considering the impact of the different step size on the prediction accuracy. Take the TC–CEEMDAN–SVR–GWO model as an example, predict each cluster that has been processed before, and study the changes in the prediction error under different prediction steps. The error results are shown in Tables 8–10. We found that, in general, the prediction error will decrease as the prediction step size increases. Specifically, the prediction error of each cluster (excluding Cluster 23) in the 3-step case is smaller than the prediction error of the cluster in the other cases. For the MAPE indicator, compared with 2-step, the error (excluding Cluster 23) under 3-step is reduced by at most 2.14% (Custer 13), at least by 0.07% (Cluster 12), and an average reduction of 1.13%. Compared with 1-step, the indicator under 3-step is reduced by 8.2% at most (Cluster 11), at least 1.4% (Cluster 24), and an average of 4.46%. Compared with 2-step, the the RMSE indicator (excluding Cluster 23) under 3-step is reduced by at most 91.19% (Cluster 21), at least by 0.69% (Cluster 24), and an average reduction of 28.14%. Compared with 1-step, the indicator under 3-step is reduced by 96.48% at most (Cluster 21), at least 24.48% (Cluster 24), and an average of 52.23%. Compared with 2-step, the the MAE indicator (excluding Cluster 23) under 3-step is reduced by at most 88.93% (Cluster 21), at least by 0.60% (Cluster 12), and an average reduction of 31.03%. Compared with 1-step, the indicator under 3-step is reduced by 96.24% at most (Cluster 21), at least 26.15% (Cluster 24), and an average of 55.29%.

  According to the above results, the prediction step size will have a certain impact on the prediction accuracy of the hybrid model. Therefore, in the process of predicting passenger flow data, the prediction step size should be set reasonably to achieve more accurate prediction.
- *different clustering conditions*: Figure 10 shows the process of using the TC–CEEMDAN–SVR–GWO model. Firstly, each sub-cluster after TC is decomposed using the CEEMDAN algorithm. The hybrid SVR–GWO model is used to predict the decomposed components, and the prediction results are integrated to obtain Cluster 1 and Cluster 2. Secondly, Cluster 1 and Cluster 2 are sorted in the correct time sequence to obtain the final short-term passenger flow prediction results.

  Compared with the result of OC and the result without clustering under 1-step, 2-step and 3-step (as shown in the Figure 11), the prediction result of TC–CEEMDAN–SVR–GWO is closer to the actual passenger flow than OC–CEEMDAN–SVR–GWO's and CEEMDAN–SVR–GWO's. At the same time, the prediction results of TC model on weekends are more accurate than those of other models and are closer to the actual passenger flow. As the number of prediction steps increases, the prediction results of each model are also more accurate. It can be seen from the Table 11 that:
  - in 1-step prediction, the prediction errors of the TC–CEEMDAN–SVR–GWO model are 8.56 (MAPE), 151.86 (RMSE) and 112.57 (MAE), respectively. Compared with OC model (OC–CEEMDAN–SVR–GWO), the prediction errors are reduced by 5.94%, 48.19% and 46.39%, respectively. Compared with the non-clustering model (CEEMDAN–SVR–GWO), the prediction errors are reduced by 5.77%, 50.9%, and 44.98%, respectively;
  - in 2-step prediction, the prediction errors of the TC–CEEMDAN–SVR–GWO model are 5.65 (MAPE), 102.28 (RMSE) and 73.01 (MAE), respectively. Compared with the OC–CEEMDAN-SVR-GWO, the prediction errors are reduced by 0.84%, 13.9%, and 8.09%, respectively. Compared with the CEEMDAN–SVR–GWO, the prediction errors were reduced by 1.07%, 11.59%, and 10.35%, respectively;
  - in 3-step prediction, the prediction errors of the TC–CEEMDAN–SVR–GWO model are 4.94 (MAPE), 82.62 (RMSE) and 62.43 (MAE), respectively. Compared with the OC–CEEMDAN–SVR–GWO, the prediction errors are reduced by 2.29%, 27.29% and 25.39%, respectively. Compared with the CEEMDAN–SVR–GWO, the prediction errors are reduced by 1.33%, 22.93%, and 17.11%, respectively.

This article considers the prediction result of CEEMDAN–BPNN–GWO model under different clustering conditions, and further illustrates the effectiveness of TC. Figure 12 shows the prediction result of the CEEMDAN–BPNN–GWO model under different clustering conditions. The prediction result of CEEMDAN–BPNN–GWO model deviates from the actual passenger flow under non-clustering conditions, which can only roughly reflect the characteristics of passenger flow fluctuations. The TC–CEEMDAN–BPNN–GWO model is more accurate in predicting the actual passenger flow, but there is still a clear deviation between the predicted result and the actual passenger flow on weekends. It can be seen from the Table 11 that:

- in 1-step prediction, the prediction errors of TC–CEEMDAN–BPNN–GWO is greatly reduced compared with OC–CEEMDAN–BPNN–GWO and CEEMDAN–BPNN–GWO, of which MAPE is reduced by 7.77% and 10.61%, respectively; RMSE is reduced by 43.23% and 47.23%, respectively; MAE decreased by 62.00% and 44.82%, respectively;
- in the 2-step prediction, compared the prediction errors of OC–CEEMDAN–BPNN–GWO and CEEMDAN–BPNN–GWO with the TC–CEEMDAN–BPNN–GWO, MAPE decreased by 2.23% and 2.52%, respectively; RMSE decreased by 12.79% and 21.49%, respectively; MAE decreased by 11.79% and 14.83%, respectively;

- in the 3-step prediction, the prediction results of TC–CEEMDAN–BPNN–GWO and OC–CEEMDAN–BPNN–GWO are similar. But compared with CEEMDAN–BPNN–GWO, TC–CEEMDAN–BPNN–GWO's, MAPE decreased by 5.03%, RMSE decreased by 23.80%, and MAE decreased by 24.57%.

Compared the prediction error results of the hybrid models under different clustering conditions (as shown in Figure 13). Under the condition of TC, the errors of 1-step prediction of all hybrid models is much smaller than that of all hybrid models under other clustering conditions. Although the prediction error values of the hybrid model under different clustering gradually approach as the number of steps increases, the prediction errors of most hybrid models under the TC are smaller than that of OC condition and the non-clustering condition. As the number of prediction steps increases, the prediction errors of the hybrid model in the case of TC also gradually become smaller. In summary, the TC method can improve the prediction accuracy of the model and ensure the accuracy of the prediction;

- *different hybrid models*: next, the performance of the hybrid SVR-GWO model proposed in this article is analyzed. Under the same clustering conditions (taking the TC condition as an example), the values of the prediction errors (MAPE, RMSE, and MAE) in the 1-step prediction can be known according to the Table 11. In the case of using the CEEMDAN decomposition algorithm, the prediction result of TC–CEEMDAN–SVR–GWO is the best (as shown in the Figure 14), and the prediction error values are 8.56, 151.86 and 121.57, respectively. They are 0.27%, 5.33% and 5.91% lower than the prediction results of TC–CEEMDAN–SVR–GA. Compared with TC–CEEMDAN–SVR, the prediction errors are lower by 0.55%, 6.85% and 8.23%, respectively. Compared with TC–CEEMDAN–BPNN–GA, the prediction errors were reduced by 0.63%, 11.22% and 9.29%, respectively.

Under the condition of EEMD decomposition (as shown in Figure 15), the prediction errors (MAPE, RMSE

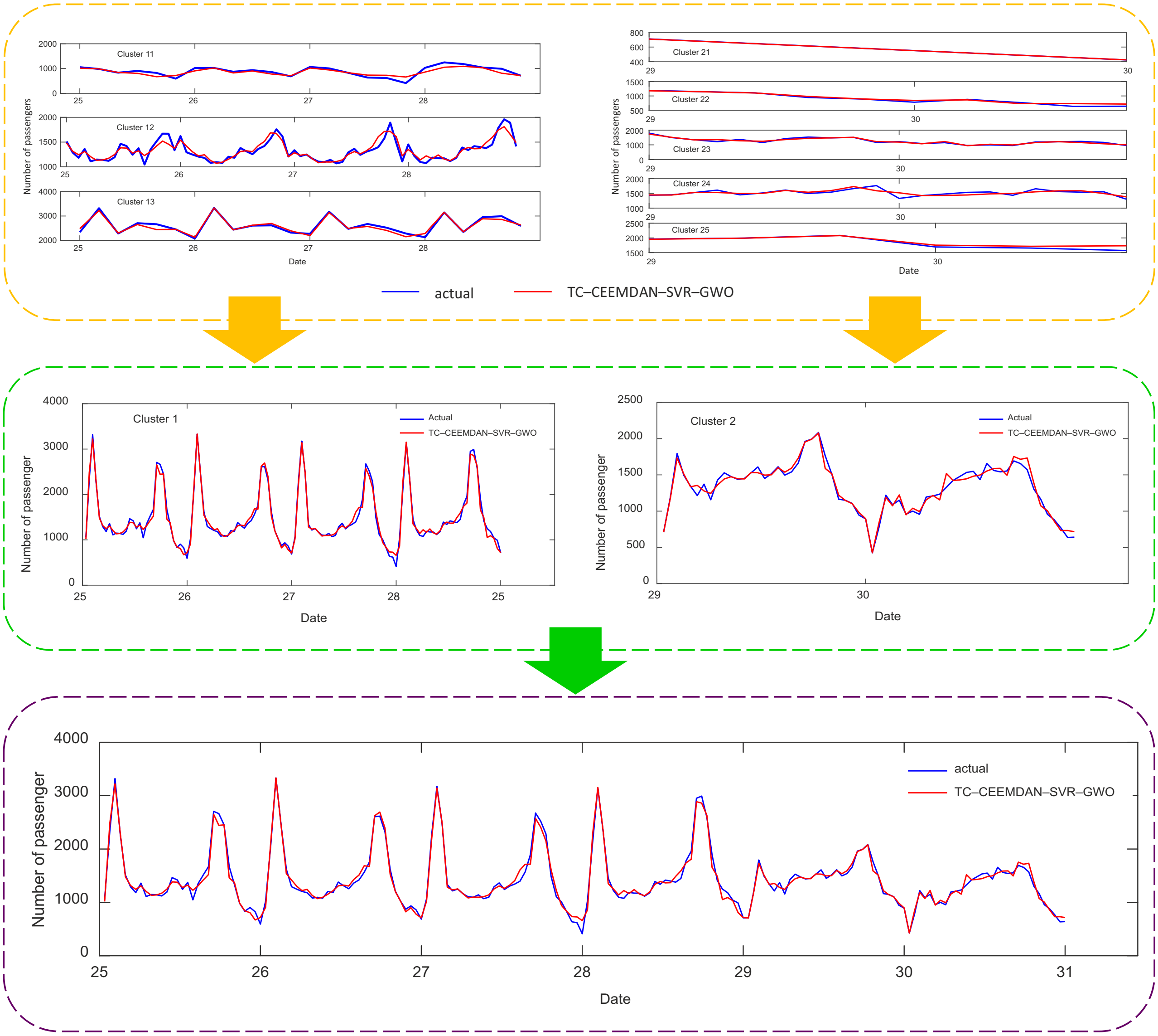

Figure 10. TC–CEEMDAN–SVR–GWO model prediction process

Table 8. The MAPE value of each cluster under different step size

Name of twice cluster	MAPE		
	1-step	2-step	3-step
Cluster 11	18.68	12.61	10.48
Cluster 12	6.86	4.84	4.77
Cluster 13	7.08	4.84	2.7
Cluster 21	5.88	1.79	0.23
Cluste r22	10.01	6.81	4.73
Cluster 23	8.14		4.05
Cluster 24	5.22	4.01	3.82
Cluster 25	7.68	4.07	3.04

and MAE) of TC–EEMD–SVR–GWO are reduced by 1.15%, 10.52% and 10.87%, respectively, compared with TC–EEMD–SVR–GA. Compared with TC–EEMD–SVR, they decreased by 0.98%, 15.31% and 11.59%, respectively. Compared with TC–EEMD–BPNN–GWO, the prediction errors are reduced by 10.9%, 76.30% and 53.59%, respectively. Compared with TC–EEMD–BPNN–GA, the prediction errors are reduced by 1.37%, 16.22%, and 14.17%, respectively.

Under the conditions of 2-step prediction and 3-step prediction, the prediction results of the hybrid SVR–GWO model under the same clustering and decomposition conditions remain optimal, and will not be repeated in this article. In summary, the novel hybrid SVR–GWO prediction model proposed in this article can achieve accurate prediction of short-term passenger flow.

Table 9. The RMSE value of each cluster under different step size

Name of twice cluster	RMSE		
	1-step	2-step	3-step
Cluster 11	193.05	121.02	106.00
Cluster 12	127.29	91.64	84.54
Cluster 13	238.86	168.84	89.19
Cluste r21	42.04	16.81	1.48
Cluster 22	115.8	62.8	46.74
Cluster 23	118.98	51.79	61.48
Cluster 24	100.61	76.51	75.98
Cluste r25	142.79	85.3	74.84

Table 10. The MAE value of each cluster under different step size

Name of twice cluster	MAE		
	1-step	2-step	3-step
Cluster 11	148.59	95.15	80.95
Cluster 12	93.15	66.25	65.85
Cluster 13	192.74	123.3	68.4
Cluster 21	36.51	12.38	1.37
Cluster 22	84.34	53.54	34.56
Cluster 23	118.98	44.46	49.86
Cluster 24	78.77	61.38	58.17
Cluster 25	136.23	68.67	49.7

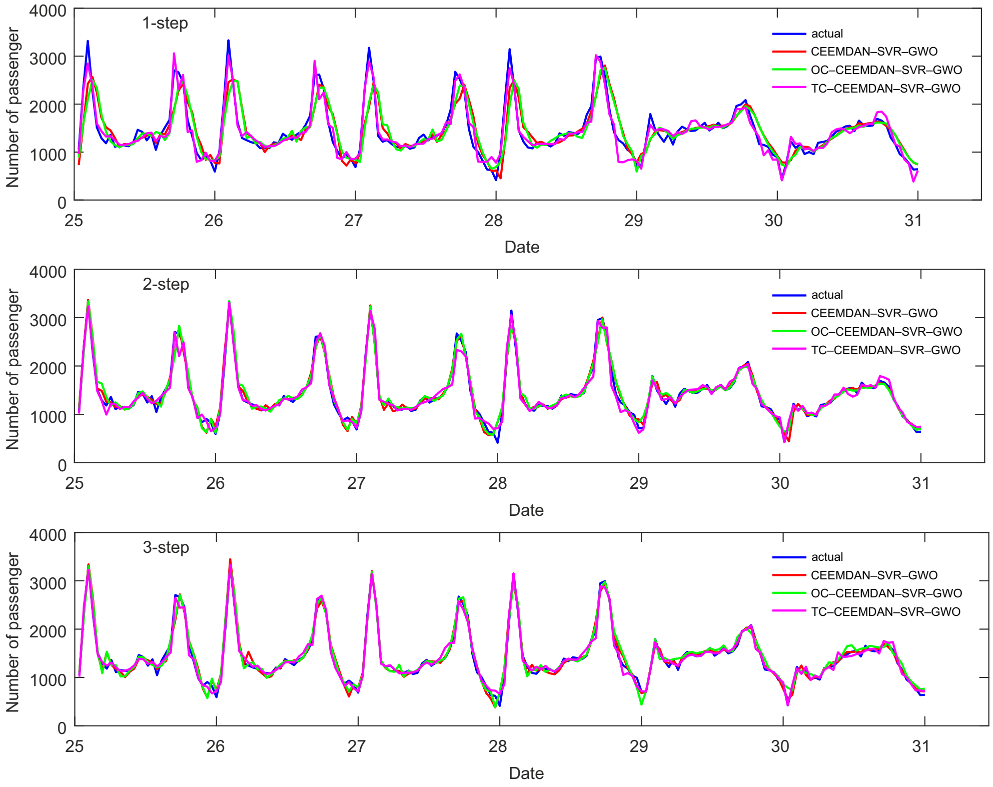

Figure 11. Multi-step prediction results of CEEMDAN–SVR–GWO model under different clustering conditions

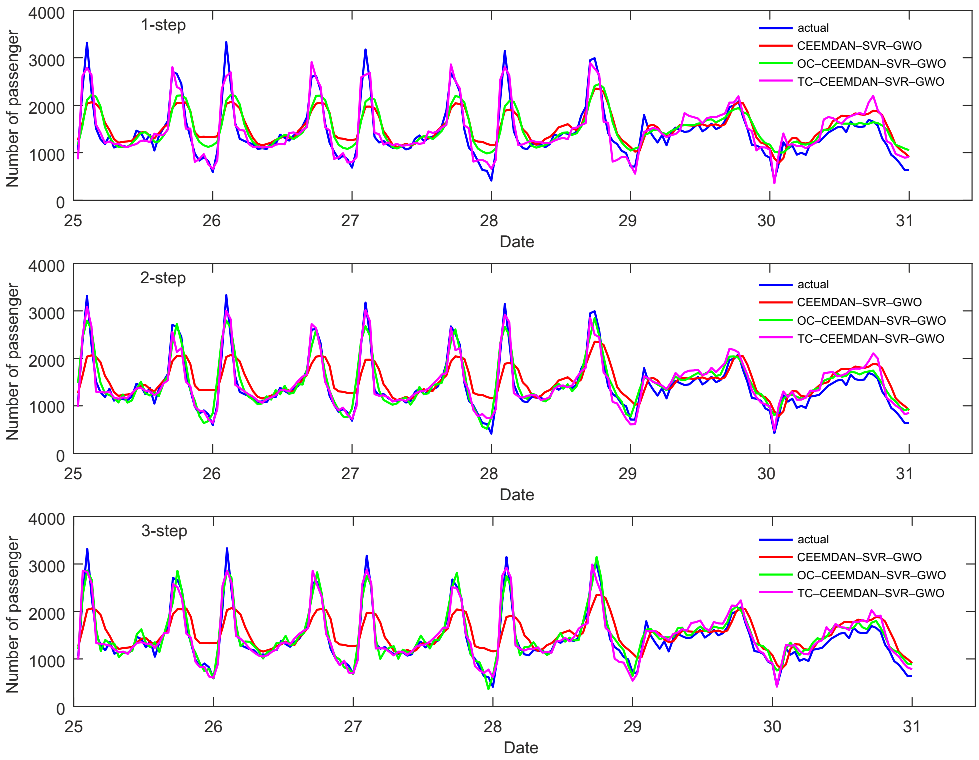

Figure 12. Multi-step prediction results of CEEMDAN–BPNN–GWO model under different clustering conditions

Table 11. Multi-step prediction error results of hybrid models in library station under different clustering conditions

Model	1-step			2-step			3-step		
	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE
TC–CEEMDAN–SVR–GWO	8.56	151.86	112.57	5.65	102.28	73.01	4.94	82.62	62.43
TC–CEEMDAN–SVR–GA	8.83	160.41	119.64	5.66	103.42	74.32	5.22	91.23	65.91
TC–CEEMDAN –SVR	9.11	163.03	122.65	7.04	142.37	98.68	6.91	136.12	95.90
TC–CEEMDAN–BPNN–GWO	11.60	199.34	156.65	9.48	164.44	128.55	9.18	168.44	128.89
TC–CEEMDAN–BPNN–GA	9.19	171.05	124.10	5.84	107.58	76.69	5.83	107.57	76.69
TC–EEMD–SVR–GWO	8.69	151.13	113.19	6.21	109.75	78.18	5.33	92.33	65.96
TC–EEMD–SVR–GA	9.84	168.90	127.00	6.23	111.25	79.71	6.01	101.49	74.60
TC–EEMD–SVR	9.67	174.27	128.03	7.97	160.64	108.68	7.71	153.00	104.89
TC–EEMD–BPNN–GWO	19.59	637.65	243.90	10.00	166.35	128.68	9.69	163.76	127.84
TC–EEMD–BPNN–GA	10.06	180.40	131.89	6.49	119.96	84.80	6.88	136.12	89.82
OC–CEEMDAN–SVR–GWO	14.50	326.6	209.98	6.49	118.79	79.44	7.23	113.63	83.68
OC–CEEMDAN–SVR–GA	15.21	332.07	217.16	6.04	117.64	75.10	7.66	136.29	97.22
OC–CEEMDAN–SVR	14.71	328.84	211.23	7.47	141.86	98.40	6.17	101.95	73.16
OC–CEEMDAN–BPNN–GWO	19.37	351.17	249.07	11.71	188.55	145.74	10.26	162.84	130.30
OC–CEEMDAN–BPNN–GA	14.63	327.40	212.86	7.74	138.43	97.24	6.64	121.86	83.48
OC–EEMD–SVR–GWO	14.70	317.21	206.47	7.95	141.50	101.04	7.45	131.70	92.99
OC–EEMD–SVR–GA	17.47	340.48	234.69	9.27	150.74	109.36	10.38	164.54	124.08
OC–EEMD–SVR	14.81	329.28	211.86	8.24	153.55	106.66	8.58	151.94	107.73
OC–EEMD–BPNN–GWO	24.11	391.29	289.99	18.25	279.14	208.71	22.05	323.07	267.00
OC–EEMD–BPNN–GA	14.67	324.67	209.16	8.56	144.59	107.91	9.04	158.24	111.78
CEEMDAN–SVR–GWO	14.33	309.31	204.62	6.72	115.69	81.44	6.27	107.20	75.32
CEEMDAN–SVR–GA	16.04	333.52	223.44	8.60	137.85	103.37	7.64	120.38	92.41
CEEMDAN–SVR	14.76	327.72	210.78	8.08	146.73	103.96	7.50	131.19	94.84
CEEMDAN–BPNN–GWO	22.21	377.72	283.87	12.00	209.45	150.93	14.21	221.04	170.88
CEEMDAN–BPNN–GA	14.46	328.76	209.49	8.16	138.65	101.46	7.49	128.28	91.43
EEMD–SVR–GWO	14.94	327.35	210.39	7.91	133.49	94.62	12.41	185.24	149.34
EEMD–SVR–GA	20.65	362.93	267.10	15.11	215.63	176.87	10.82	168.40	132.67
EEMD–SVR	15.22	329.25	214.10	9.07	157.69	113.24	11.72	180.90	142.87
EEMD–BPNN–GWO	30.44	443.10	357.81	22.54	322.74	259.73	13.87	214.55	167.58
EEMD–BPNN–GA	16.68	331.93	223.31	8.36	140.88	102.79	12.41	185.24	149.34

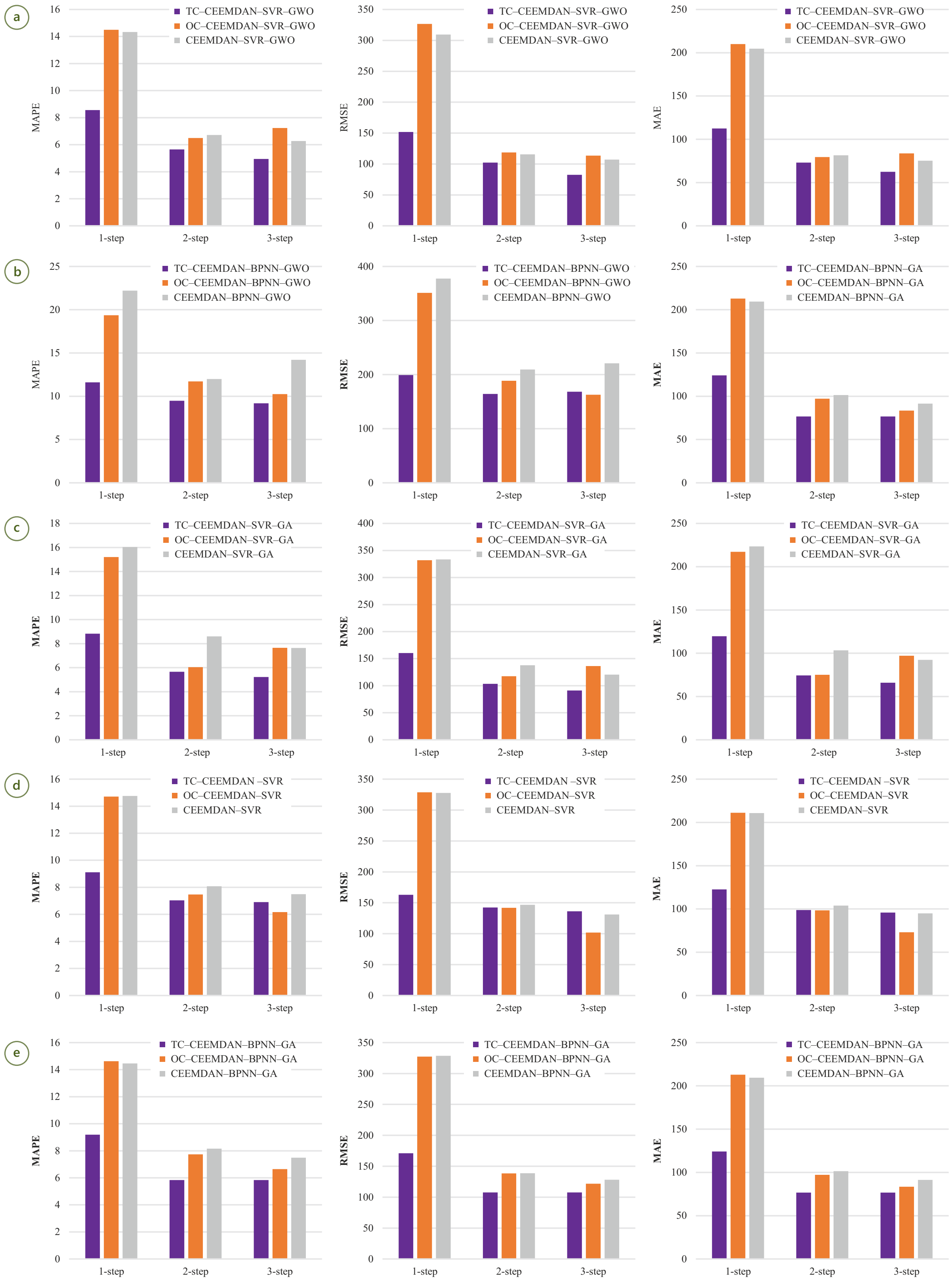

Figure 13. Comparison of prediction errors under different clustering conditions:
a – comparison of CEEMDAN–SVR–GWO model prediction errors under TC, OC, and no clustering conditions;
b – comparison of CEEMDAN–BPNN–GWO model prediction errors under TC, OC, and no clustering conditions;
c – comparison of CEEMDAN–SVR–GA model prediction errors under TC, OC, and no clustering conditions;
d – comparison of CEEMDAN–SVR model prediction errors under TC, OC, and no clustering conditions;
e – comparison of CEEMDAN–BPNN–GA model prediction errors under TC, OC, and no clustering conditions

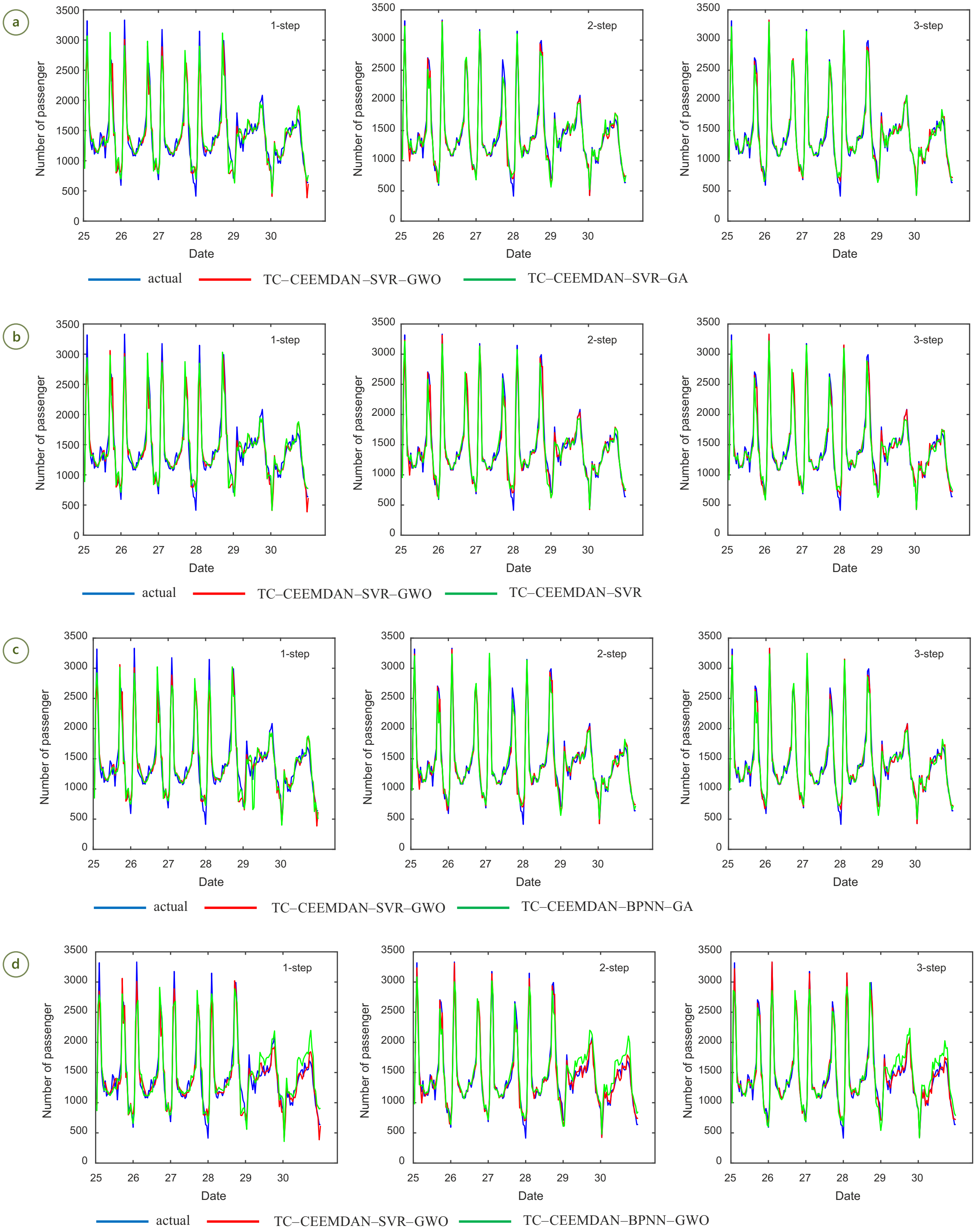

Figure 14. Comparison of different hybrid models using CEEMDAN algorithm under TC:

a – comparison of SVR-GWO model and SVR-GA model using CEEMDAN algorithm under TC in 1-step, 2-step and 3-step;

b – comparison of SVR-GWO model and SVR model using CEEMDAN algorithm under TC in 1-step, 2-step and 3-step;

c – comparison of SVR-GWO model and BPNN-GA model using CEEMDAN algorithm under TC in 1-step, 2-step and 3-step;

d – comparison of SVR-GWO model and BPNN-GWO model using CEEMDAN algorithm under TC in 1-step, 2-step and 3-step

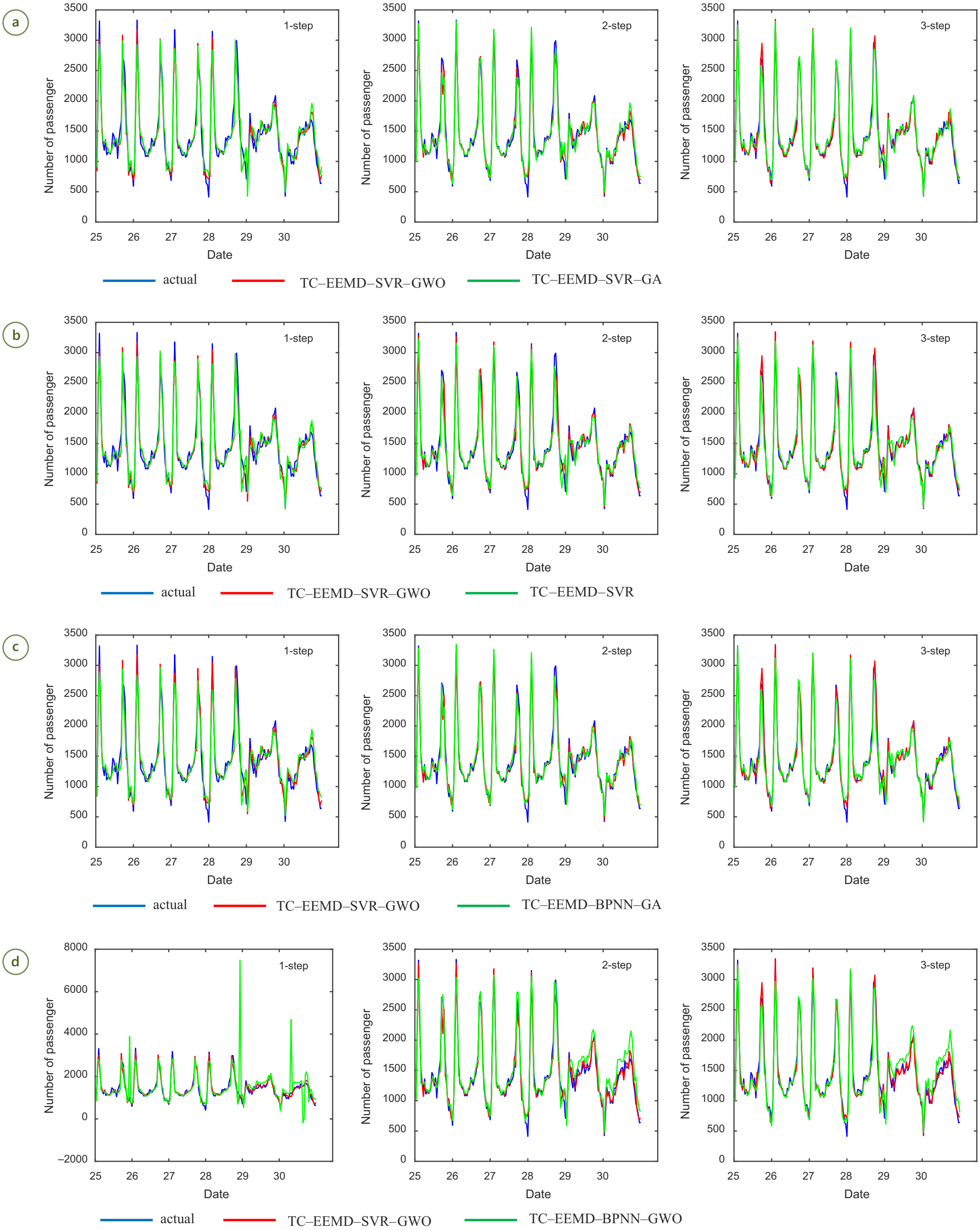

Figure 15. Comparison of different hybrid models using EEMD algorithm under TC:

a – comparison of SVR-GWO model and SVR-GA model using EEMD algorithm under TC in 1-step, 2-step and 3-step;

b – comparison of SVR-GWO model and SVR model using EEMD algorithm under TC in 1-step, 2-step and 3-step;

c – comparison of SVR-GWO model and BPNN-GA model using EEMD algorithm under TC in 1-step, 2-step and 3-step;

d – comparison of SVR-GWO model and BPNN-GWO model using EEMD algorithm under TC in 1-step, 2-step and 3-step

5. Conclusions

Accurate forecasting model can help transportation system operations. This article proposes a novel hybrid SVR–GWO model based on TC to achieve short-term passenger flow prediction for urban rail transit. The major steps of the method proposed in this article are as follows:

- collecting the original passenger flow data and using the *k*-means method to achieve TC of the data;
- a series of sub-cluster are obtained through TC, and each sub-clusters is decomposed using CEEMDAN algorithm or EEMD algorithm;
- using the hybrid SVR–GWO model to predict the decomposed IMF components;
- assemble the predicted data to get the final prediction result.

TC can effectively adequately model the volatility and make each final prediction result more accurate. Comparing the prediction results of the same hybrid model under different clustering conditions, the effectiveness of the TC method in short-term passenger flow prediction is illustrated. At the same time, adding the GWO algorithm to the SVR-based model can quickly and effectively optimize the parameters of the SVR model and improve the prediction accuracy of the hybrid model. And compared with SVR–GA, BPNN–GWO and other different models, it further proved the accuracy of SVR–GWO prediction. The results show that the novel hybrid TC–CEEMDAN–SVR–GWO model can effectively reduce prediction errors and achieve accurate prediction, which will be useful work in the future.

Accurate short-term urban rail transit passenger flow forecast can more effectively play the capacity of urban rail transit equipment. At the same time, accurate prediction has strong practical significance in rationally optimizing the operation of urban rail transit lines, ensuring the safety and speed of passenger travel, and reducing the cost of operating departments.

Therefore, the importance of accurate passenger flow forecasting lies in alleviating the pressure on the station, evacuating passengers at the station in time, realizing reasonable planning, and ensuring the safety of residents.

Although this article performs well in short-term forecasting, it only considers the impact of passenger flow fluctuations on passenger flow forecasting. It does not consider restrictions such as weather conditions, ticketing and ticket checking equipment capacity of different stations and the impact of factors such as sudden large passenger flow caused by competition activities. It will make the predicted results more accurate if considering the above factors. Therefore, we will further explore the above issues in future research.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No 71761024).

Author contributions

Sheng Wang and *Xinfeng Yang* conceived and designed the study.

Sheng Wang completed model analysis.

Sheng Wang and *Xinfeng Yang* wrote the article.

All authors performed the experiments; reviewed, edited and approved the article.

Disclosure statement

This article do not have any competing financial, professional, or personal interests from other parties.

References

Cortes, C.; Vapnik, V. 1995. Support-vector networks, *Machine Learning* 20(3): 273–297. https://doi.org/10.1007/BF00994018

CURTA. 2019. *Urban Rail Transit 2018 Statistics and Analysis Report*. China Urban Rail Transit Association (CURTA), China. (in Chinese).

Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function, *Mathematics of Control, Signals and Systems* 2(4): 303–314. https://doi.org/10.1007/BF02551274

Deng, Y.; Xiang, J.; Ou, Z. 2012. SVR with hybrid chaotic genetic algorithm for short-term traffic flow forecasting, in *2012 8th International Conference on Natural Computation*, 29–31 May 2012, Chongqing, China, 708–712. https://doi.org/10.1109/ICNC.2012.6234768

Hamed, M. M.; Al-Masaeid, H. R.; Said, Z. M. B. 1995. Short-term prediction of traffic volume in urban arterials, *Journal of Transportation Engineering* 121(3): 249–254. https://doi.org/10.1061/(ASCE)0733-947X(1995)121:3(249)

Hao, S.; Lee, D.-H.; Zhao, D. 2019. Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system, *Transportation Research Part C: Emerging Technologies* 107: 287–300. https://doi.org/10.1016/j.trc.2019.08.005

Heydari, A.; Astiaso Garcia, D.; Keynia, F.; Bisegna, F.; De Santoli, L. 2019. Renewable energies generation and carbon dioxide emission forecasting in microgrids and national grids using GRNN-GWO methodology, *Energy Procedia* 159: 154–159. https://doi.org/10.1016/j.egypro.2018.12.044

Hochreiter, S.; Schmidhuber, J. 1997. Long short-term memory, *Neural Computation* 9(8): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hong, W.-C. 2011. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm, *Neurocomputing* 74(12–13): 2096–2107. https://doi.org/10.1016/j.neucom.2010.12.032

Hu, Y.; Wu, C.; Liu, H. 2011. Prediction of passenger flow on the highway based on the least square support vector machine, *Transport* 26(2): 197–203. https://doi.org/10.3846/16484142.2011.593121

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. 1998. The empirical mode decomposition and the Hilbert spectrum for nonlinear and nonstationary time series analysis, *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences* 454(1971): 903–995. https://doi.org/10.1098/rspa.1998.0193

Kamarianakis, Y, Prastacos, P. 2005. Space–time modeling of traffic flow, *Computers & Geosciences* 31(2): 119–133. https://doi.org/10.1016/j.cageo.2004.05.012

Kumar, S. V.; Vanajakshi, L. 2015. Short-term traffic flow prediction using seasonal ARIMA model with limited input data, *European Transport Research Review* 7: 21. https://doi.org/10.1007/s12544-015-0170-8

Lee, S.; Fambro, D. B. 1999. Application of subset autoregressive integrated moving average model for short-term freeway traffic volume forecasting, *Transportation Research Record*: *Journal of the Transportation Research Board* 1678: 179–188. https://doi.org/10.3141/1678-22

Lei, Y.; Lin, J.; He, Z; Zuo, M. J. 2013. A review on empirical mode decomposition in fault diagnosis of rotating machinery, *Mechanical Systems and Signal Processing* 35(1–2): 108–126. https://doi.org/10.1016/j.ymssp.2012.09.015

Li, H; Liu, T.; Wu, X.; Chen, Q. 2019a. Application of EEMD and improved frequency band entropy in bearing fault feature extraction, *ISA Transactions* 88: 170–185. https://doi.org/10.1016/j.isatra.2018.12.002

Li, H.; Wang, Y.; Xu, X.; Qin, L.; Zhang, H. 2019b. Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network, *Applied Soft Computing* 83: 105620. https://doi.org/10.1016/j.asoc.2019.105620

Li, M.-W.; Hong, W.-C.; Kang, H.-G. 2013. Urban traffic flow forecasting using Gauss–SVR with cat mapping, cloud model and PSO hybrid algorithm, *Neurocomputing* 99: 230–240. https://doi.org/10.1016/j.neucom.2012.08.002

Liu, L.; Chen, R.-C. 2017. A novel passenger flow prediction model using deep learning methods, *Transportation Research Part C*: *Emerging Technologies* 84: 74–91. https://doi.org/10.1016/j.trc.2017.08.001

Ma, X.; Tao, Z.; Wang, Yu.; Yu, H.; Wang, Yu. 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, *Transportation Research Part C*: *Emerging Technologies* 54: 187–197. https://doi.org/10.1016/j.trc.2015.03.014

Mirjalili, S.; Mirjalili, S. M.; Lewis, A. 2014. Grey wolf optimizer, *Advances in Engineering Software* 69: 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Mrówczyńska, B.; Łachacz, K.; Haniszewski, T.; Sładkowski, A. 2012. A comparison of forecasting the results of road transportation needs, *Transport* 27(1): 73–78. https://doi.org/10.3846/16484142.2012.666763

Niu, M.; Wang, Y.; Sun, S.; Li, Y. 2016. A novel hybrid decomposition-and-ensemble model based on CEEMD and GWO for short-term $PM_{2.5}$ concentration forecasting, *Atmospheric Environment* 134: 168–180. https://doi.org/10.1016/j.atmosenv.2016.03.056

Plakandaras, V.; Papadimitriou, T.; Gogas, P. 2019. Forecasting transportation demand for the U.S. market, *Transportation Research Part A*: *Policy and Practice* 126: 195–214. https://doi.org/10.1016/j.tra.2019.06.008

Ramakrishnan, T.; Sankaragomathi, B. 2017. A professional estimate on the computed tomography brain tumor images using SVM-SMO for classification and MRG-GWO for segmentation, *Pattern Recognition Letters* 94: 163–171. https://doi.org/10.1016/j.patrec.2017.03.026

Ren, G.; Zhou, Z. 2011. Traffic safety forecasting method by particle swarm optimization and support vector machine, *Expert Systems with Applications* 38(8): 10420–10424. https://doi.org/10.1016/j.eswa.2011.02.066

Rumelhart, D. E.; Hinton, G. E.; Williams, R. J. 1986. Learning representations by back-propagating errors, *Nature* 323: 533–536. https://doi.org/10.1038/323533a0

Schölkopf, B.; Smola, A. J. 2001. *Learning with Kernels*: *Support Vector Machines*, *Regularization*, *Optimization*, *and Beyond*. MIT Press. 644 p. https://doi.org/10.7551/mitpress/4175.001.0001

Sulaiman, M. H.; Mustaffa, Z.; Mohamed, M. R.; Aliman, O. 2015. Using the gray wolf optimizer for solving optimal reactive power dispatch problem, *Applied Soft Computing* 32: 286–292. https://doi.org/10.1016/j.asoc.2015.03.041

Šliupas, T. 2006. Annual average daily traffic forecasting using different techniques, *Transport* 21(1): 38–43. https://doi.org/10.3846/16484142.2006.9638039

Van Der Voort, M.; Dougherty, M.; Watson, S. 1996. Combining Kohonen maps with ARIMA time series models to forecast traffic flow, *Transportation Research Part C*: *Emerging Technologies* 4(5): 307–318. https://doi.org/10.1016/S0968-090X(97)82903-8

Vlahogianni, E. I.; Karlaftis, M. G.; Golias, J. C. 2014. Short-term traffic forecasting: where we are and where we're going, *Transportation Research Part C*: *Emerging Technologies* 43: 3–19. https://doi.org/10.1016/j.trc.2014.01.005

Wang, J.; Shi, Q. 2013. Short-term traffic speed forecasting hybrid model based on chaos–wavelet analysis-support vector machine theory, *Transportation Research Part C*: *Emerging Technologies* 27: 219–232. https://doi.org/10.1016/j.trc.2012.08.004

Wang, K.; Qi, X.; Liu, H.; Song, J. 2018. Deep belief network based *k*-means cluster approach for short-term wind power forecasting, *Energy* 165: 840–852. https://doi.org/10.1016/j.energy.2018.09.118

Williams, B M. 2001. Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling, *Transportation Research Record*: *Journal of the Transportation Research Board* 1776: 194–200. https://doi.org/10.3141/1776-25

Wu, Y.-X.; Wu, Q.-B.; Zhu, J.-Q. 2019. Improved EEMD-based crude oil price forecasting using LSTM networks, *Physica A*: *Statistical Mechanics and its Applications* 516: 114–124. https://doi.org/10.1016/j.physa.2018.09.120

Wu, Z.; Huang, N. E. 2009. Ensemble empirical mode decomposition: a noise-assisted data analysis method, *Advances in Adaptive Data Analysis* 1(1): 1–41. https://doi.org/10.1142/S1793536909000047

Yang, H.-F.; Dillon, T. S.; Chen, Y.-P. P. 2017. Optimized structure of the traffic flow forecasting model with a deep learning approach, *IEEE Transactions on Neural Networks and Learning Systems* 28(10): 2371–2381. https://doi.org/10.1109/TNNLS.2016.2574840

Yang, X.; Xue, Q.; Ding, M.; Wu, J.; Gao, Z. 2021a. Short-term prediction of passenger volume for urban rail systems: a deep learning approach based on smart-card data, *International Journal of Production Economics* 231: 107920. https://doi.org/10.1016/j.ijpe.2020.107920

Yang, X.; Xue, Q.; Yang, X.-X.; Yin, H.; Qu, Y.; Li, X.; Wu, J. 2021b. A novel prediction model for the inbound passenger flow of urban rail transit, *Information Sciences* 566: 347–363. https://doi.org/10.1016/j.ins.2021.02.036

Zhang, G. P. 2003. Time series forecasting using a hybrid ARIMA and neural network model, *Neurocomputing* 50: 159–175. https://doi.org/10.1016/S0925-2312(01)00702-0

Zhang, G.; Patuwo, B. E.; Hu, M. Y. 1998. Forecasting with artificial neural networks: the state of the art, *International Journal of Forecasting* 14(1): 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7

Zhang, H.; Wang, X.; Cao, J.; Tang, M.; Guo, Y. 2018. A hybrid short-term traffic flow forecasting model based on time series multifractal characteristics, *Applied Intelligence* 48(8): 2429–2440. https://doi.org/10.1007/s10489-017-1095-9

Zhong, C.; Batty, M.; Manley, E.; Wang, J.; Wang, Z.; Chen, F.; Schmitt, G. 2016. Variability in regularity: mining temporal mobility patterns in London, Singapore and Beijing using smart-card data, *PLoS ONE* 11(2): e0149222. https://doi.org/10.1371/journal.pone.0149222

Zhu, S.; Qiu, X.; Yin, Y.; Fang, M.; Liu, X.; Zhao, X.; Shi, Y. 2019. Two-step-hybrid model based on data preprocessing and intelligent optimization algorithms (CS and GWO) for NO_2 and SO_2 forecasting, *Atmospheric Pollution Research* 10(4): 1326–1335. https://doi.org/10.1016/j.apr.2019.03.004