

## Research Article

# A Novel Hierarchical Hybrid Model for Short-Term Bus Passenger Flow Forecasting

Huawei Zhai <sup>1,2</sup>, Ruijie Tian,<sup>1</sup> Licheng Cui <sup>3</sup>, Xiaowei Xu <sup>2</sup> and Weishi Zhang<sup>1</sup>

<sup>1</sup>Information Science and Technology School, Dalian Maritime University, Dalian 116026, China

<sup>2</sup>University of Arkansas at Little Rock, Little Rock, AR, USA

<sup>3</sup>Public Security Information Department, Liaoning Police College, Dalian 116036, China

Correspondence should be addressed to Licheng Cui; [clclcheng@163.com](mailto:clclcheng@163.com) and Xiaowei Xu; [1xiaowei.xu@gmail.com](mailto:1xiaowei.xu@gmail.com)

Received 15 June 2019; Revised 9 September 2019; Accepted 15 November 2019; Published 1 February 2020

Academic Editor: Rocío de Oña

Copyright © 2020 Huawei Zhai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

For the increasing travel demands and public transport problems, dynamically adjusting timetable or bus scheduling is necessary based on accurate real-time passenger flow forecasting. In order to get more accurate passenger flow in future, this paper proposes a novel hierarchical hybrid model based on time series model, deep belief networks (DBNs), and improved incremental extreme learning machine (Im-ELM) to forecast short-term passenger flow. The proposed model is named HTSDBNE with two modelling steps. First, referring the idea of parallelization, the hybrid model, constructed by time series model, DBN, and Im-ELM, is used to forecast short-term passenger flow in different time scales hierarchically and parallel. Second, Im-ELM is utilized to analyse the relationship of forecasting results from the first step, and the weighted outputs of Im-ELM are as the final forecasting results. Comparing with single forecasting models and typical hybrid forecasting models, the testing results indicate that HTSDBNE has better performances. The mean absolute percent error of prediction results is around 10% and fully meets the application requirements of bus operation enterprise.

## 1. Introduction

For current urban bus transport system, it faces more and more problems, such as improper arriving of buses, overcrowded or empty carriages, and so on, which cause passengers delay, bad ride experiences, and waste of transport resources. Thus, many enterprises try to adopt dynamically setting the timetable in real time based on the passenger flow variations and provide services in a proactive manner as opposed to a reactive manner with a predictive capability [1, 2]. Short-term passenger flow prediction, the forecasting time interval not exceeding 60 minutes, is essentially important for setting the timetable in real time. It is one of the most significant basics for the operation planning and decision making so as to rationally utilize transport resources, solve or ease transport problems, and provide better bus services [3, 4].

In recent decades, short-term passenger flow prediction has drawn the widespread attention, and various methods

have been proposed, which could be categorized as linear models, nonlinear models, and combination models generally [5]. Because the passenger flow statistics are naturally time dependent, the linear models, such as autoregressive integrated moving average (ARIMA), autoregressive moving average (ARMA), and autoregressive (AR) models, are widely used for simple short-term passenger flow prediction. Ma et al. [6] and Xue et al. [7] constructed a combined forecasting model based on multiple time series algorithms to forecast the changes of passenger flow in different time periods. However, linear models are limited in applications and hard to describe the variation characteristics of passenger flow. For tracking the nonlinear characteristics of real passenger flow, many nonlinear methods have been introduced by researchers, such as the support vector machine (SVM) model [8], least squares support vector machine (LSSVM) model [9], fuzzy neural networks [10, 11], Bayesian network [12, 13], radial basis function neural networks (RBF-ANN) [14–17], and grey model [18–20]. The

core idea of these nonlinear methods is to construct the nonlinear relationship between passenger flow and mine more potential information without prior knowledge [21]. However, the nonlinear models are closely related to the specific application environment and lack universality.

Each method mentioned above has its own advantages and limitations in real applications, and it is hard to cover all characteristics of passenger flow and provide the best prediction performances globally by a single method [6]. Thus, the hybrid prediction model, combining multiple algorithms strategically, can make full use of the advantages of different algorithms and cover the shortcomings of each algorithm, and it has become common practices in improving the prediction performances. For example, Sun et al. [22] proposed a novel hybrid model Wavelet-SVM; it utilized SVM model to capture passenger flow characteristics from different frequencies generated by wavelet decomposition. Yang and Liu [8] introduced affinity propagation to cluster the passenger flow based on the characteristic analysis and then utilized SVM to predict each subset; the prediction accuracy was improved significantly. Liu et al. [23] proposed a combinational prediction model BP-LSSVM; the initial prediction results from BP are refined by LSSVM further to obtain better predicted passenger volume. Wang et al. [24] used Levenberg–Marquardt algorithm to optimize the BP, and constructed SLMBP model with Spearman rank correlation coefficient method [25] to predict passenger flow. Among these hybrid methods, the ANN-based model has been utilized widely because of its better predictive ability [26].

Furthermore, in recent years, deep learning (DL) has attracted considerable academic and industrial interests [27]. Some DL-based hybrid models have been applied in passenger flow prediction, as they could represent the complex nonlinear relationship and capture the latent correlative features from passenger flow data. Liu et al. [27] proposed an end-to-end DL architecture, for short-term metro passenger flow prediction. Bai et al. [21] presented a multipattern deep fusion approach (MPDF), which is constructed by fusing deep belief networks (DBNs) corresponding to multiple patterns, and it utilized DBNs as a deep representation for passenger flow in each pattern generated by the affinity propagation algorithm. Ke et al. [28] proposed a novel DL approach, named the fusion convolutional long short-term memory network. It is stacked and fused by multiple convolutional long short-term memory (LSTM) layers, standard LSTM layers, and convolutional layers, which capture the spatiotemporal correlations of passenger demands accurately. Liu et al. [29] presented a passenger hybrid estimation system based on the convolutional neural network (CNN) and the spatiotemporal context (STC) model. CNN is used to detect the passengers, and then STC is used to track the passengers so as to accurately estimate the passenger volume.

In summary, the linear model is simple in structure, the accuracy of nonlinear algorithms is better than linear ones, and the combination model is more adaptable. However, these research results mainly include the following defects. (1) Data problem: the samples of passenger flow statistics studied in most research studies are all from the automatic

fare collection (AFC) system [30], which cannot cover the number of passengers who buy tickets in cash. According to the equipment currently used, most AFC systems of bus cannot transmit ticket information to the bus operation enterprise in real time, and in this case, the data samples cannot be used for real time prediction. (2) Passenger flow is complex time series data; it has its own special microscopic characteristic and macroscopic characteristics, and most of studies do not analyse in this aspect; they only consider global characteristics or only analyse the nonlinear nature. (3) Passenger flow in different time scales is correlative with each other. However, many research studies only assume a linear relation between their patterns [31, 32], which could lead to underestimate or degenerate performances, and most hybrid models are too complex to be used in practices.

In order to solve the problems mentioned above, make full use of the advantages of linear and nonlinear models, improve the universality and accuracy of the models, and reduce model complexity, this paper proposes a short-term passenger flow hierarchical hybrid forecasting model based on time series model, DBN and Im-ELM, called HTSDBNE. The real-time passenger flow data collected by the automatic passenger counting (APC) [25] systems are selected as statistical samples for forecasting. HTSDBNE finishes the forecasting operation by two steps: (a) utilizing the time series model and the subhybrid model, consisting of DBN [33, 34] and Im-ELM [35, 36], to analyse the statistical data and forecast the variation trends of the passenger flow and (b) analysing the relationship between the real-time and historical passenger flow and make full use of passenger flow series data in different time scales to improve the final result accuracy.

The rest of the paper is organized as follows. Section 2 describes the structure of the bus passenger flow sampling data. Section 3 discusses the novel hybrid forecasting model. Section 4 shows the comparative experiments and analysis. At last, Section 5 summarizes the main findings and future work.

## 2. Bus Passenger Flow Statistics

*2.1. Data Source.* There are mainly three approaches to obtain bus passenger flow statistics: one is survey on buses or at stops manually. The other two statistical methods are through the AFC and APC systems. Manual statistics are now almost no longer used due to low efficiency and high cost. Because the AFC system is widely installed on buses, passenger flow can be inferred from passenger ticket information, and it has become the main source of passenger flow statistics. However, in the current bus system, a considerable number of passengers use cash to buy tickets, so the passenger flow statistic results from the AFC system cannot fully cover all passengers. The APC system is able to conduct passenger flow statistics relatively comprehensively and accurately and has become one of the important development directions for bus passenger flow statistics. The passenger flow statistics used in this paper are from the APC system, and the structure of the record is shown in Table 1 [5].

**2.2. Dataset Definition.** The original passenger flow statistics set generated by the APC installed on the bus is related to the bus outbound time. This paper defines the relevant dataset as follows.

Define the arrival bus sequence at a station as

$$\text{BUS} := \{\text{busID}_i, i = 1, 2, \dots, n\}, \quad (1)$$

where  $\text{busID}_1$  is the first bus arriving at the station in one day and  $\text{busID}_n$  is the last bus arriving at the station in one day.

Define the passenger flow statistics sequence of a stop in a day as an ordered list:

$$\begin{aligned} PC^{\text{day}} &:= \{Pc_i^d, i = 1, 2, \dots, n\}, \\ Pc_i^d &:= \langle \text{stopID}, \text{busID}_i, t_i, \text{count} \rangle, \end{aligned} \quad (2)$$

where  $\text{stopID}$  is the number of the bus stop and  $t_i$  is the time when passenger flow statistics are uploaded to the database in real time after the bus with  $\text{busID}_i$  leaves the stop. The count is the sum of passengers getting on the bus with  $\text{busID}_i$  at a stop with  $\text{stopID}$ .

**2.3. Data Sample Analysis.** In this paper, the passenger flow statistics used are derived from the APC system installed on the buses of line 28 and line 10 in Dalian, China. Some stops with large passenger flow volume determine the whole line's passenger flow variation. Therefore, in related research studies, the bus stops with relatively larger passenger flow are usually selected as research objects.

Firstly, the daily average passenger flow of each station in the up direction of line 28 in the past six months was counted. The result is shown in Figure 1.

As shown in Figure 1, the origin station (station 1), station 3, and station 7 had an average daily ridership of over 1000. These stations are important stops in the upward direction of the line 28. In this paper, station 7 is selected as the research object for short-term passenger flow forecasting.

The sample data from station 7 used in this paper are 26 weeks' statistics from Monday 1 October 2018 to Sunday 31 March 2019. The part of origin data diagram is shown in Figure 2.

As shown in Figure 2, the original passenger flow statistics are related to the arrival time of the buses. Due to traffic congestion or other reasons, it is difficult for each bus to arrive at the stop on time according to the timetable, which causes the data sample intervals to be not equal. In order to reduce the instability of the passenger flow statistical sequence caused by abnormal factors, the data are aggregated in equal time intervals to construct a time series as shown below.

Define time interval as

$$\text{time}_{\text{interval}} := \left\{ \text{time}_{\text{interval}} \mid \text{time}_{\text{interval}} \in (1 \text{ min}, 2 \text{ min}, \dots, 60 \text{ min}) \right\}. \quad (3)$$

Generally, time interval is determined by bus scheduling plan, which is not less than the minimum departure interval. In the applied research of short-term passenger flow

forecasting, the maximum time interval cannot exceed 60 minutes, and the equal time interval of 5, 15, and 30 minutes is selected in this paper.

Data segmentation time point can be determined according to the time interval, defined as

$$\text{time}_l := \tilde{t}_1 + l * \text{time}_{\text{interval}}, \quad l = 1, 2, \dots, \max(l), \quad (4)$$

where  $\tilde{t}_1$  is the earliest time of the first bus leaving the stop in all statistics. Define  $\max(l)$  as

$$\max(l) = \frac{(\tilde{t}_n - \tilde{t}_1) * 60}{\text{time}_{\text{interval}}}, \quad (5)$$

where  $\tilde{t}_n$  is the latest time for the last bus leaving the stop.

The new passenger flow statistical sequence formed after equal time interval convergence is defined as follows:

$$\overline{PC}^{\text{day}} := \left\{ \overline{Pc}_k^d, k = 1, 2, \dots, \max(l) \right\}, \quad (6)$$

$$\overline{Pc}_k^d := \langle \text{stopID}, \text{time}_{e_k}, \text{tcount} \rangle, \quad (7)$$

$$\overline{Pc}_k^d \cdot \text{tcount} = \sum_{i=p}^q Pc_i^d \cdot \text{count}, \quad (8)$$

$$\text{time}_{e_q} < \text{time}_{e_1} + q * \text{time}_{\text{interval}}, \quad (9)$$

$$\text{time}_{e_p} \geq \text{time}_{e_1} + p * \text{time}_{\text{interval}}. \quad (10)$$

According to equation (7), the passenger flow statistics of station 7 are aggregated in the interval of 30 minutes, and the statistical results are shown in Figure 3.

It can be concluded from the curve changes that daily passenger flow shows double-peak changes in the morning and evening. The early peak period of the working day can be from 7:30 to 9:30, and the evening peak period can be from 16:30 to 18:30. In the following study, the model proposed in this paper will be used to forecast the changes of passenger flow during the morning peak period, and the datasets are aggregated in the interval of 30 minutes. From the observation in Figure 3, the passenger flow statistical sequence has time-periodic variation characteristics with linear correlation. However, the changes in each cycle are not exactly the same, with obvious nonlinear characteristics. Therefore, it is necessary to combine linear and nonlinear methods to describe the passenger flow statistical sequence in order to accurately forecast the passenger flow variation.

### 3. Hybrid Forecasting Model

The theoretical and empirical findings have already indicated that the integrated model of different models is an effective way for improving the forecasting performances and making up for the shortcomings of each model. The proposed hybrid forecasting model is based on the previous works. Khashei and Bijari [37] and Zhang [32] employed linear models to combine with neural network model, by using linear model to identify and magnify the linear structure of the data and then using neural network to model

TABLE 1: The structure of the record from the automatic passenger counters (APCs).

Field name	Illustration
Equipment ID	The unique number of the equipment
On/off	Denotes the passenger getting on or off the bus
Vehicle ID	The unique number of the vehicle
Line ID	The unique number of the bus line
Trip type	Denotes the current trip is up run or down run
Stop ID	The unique number of stops, where the bus stops at the current time
Count time	The time when the passenger scans through the AFC
Stop accumulation	The total number of the passengers getting on or off at a stop

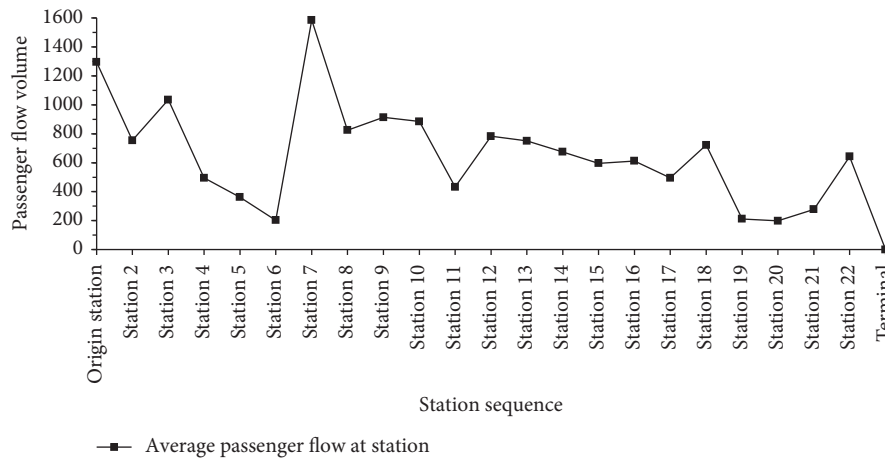


FIGURE 1: Daily average passenger flow at all stations of bus line 28.

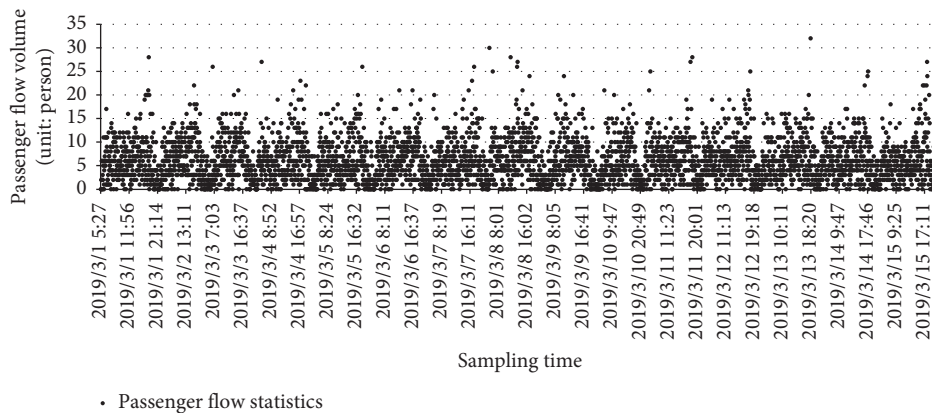


FIGURE 2: Original passenger flow statistics of station 7 of line 28 within 15 days.

the preprocessed data in order to improve the prediction accuracy. Some works [32, 37–40] considered the importance of the residual series of time series data and combined the time series forecasting results to improve the performances of the hybrid model. After referring these works and some idea of online sequential algorithms, a novel hybrid forecasting model is proposed.

In general, the novel model performs data modelling in two sequential steps. One is hybrid time series modelling, which is used to analyse the time series from linear and nonlinear characteristic. The other is nonlinear data modelling, and it is used to analyse the previous results from different time scales or time and spatial scales.

Figure 4 shows the process of the first step; in general, it performs in three sequential substeps. (I) Using the linear model to forecast the time series data. Given the training time series set  $X_t$ , the forecasting result is  $X_{t,L}$ . For the instability of passenger flow statistical series, the result  $X_{t,L}$  is not stable and unacceptable, and it cannot be used as the final results. (II) Analysing and forecasting the residual series. Based on the real value  $X_t$  and time series forecasting result  $X_{t,L}$ , the error or residual is calculated from their difference,  $E_t = X_t - X_{t,L}$ , and then the residual series is used to train the nonlinear model, whose output is  $X_{t,NL}$ . The nonlinear model is a hybrid model, consisting of DBN and Im-ELM, and the Im-ELM is used to forecast residual

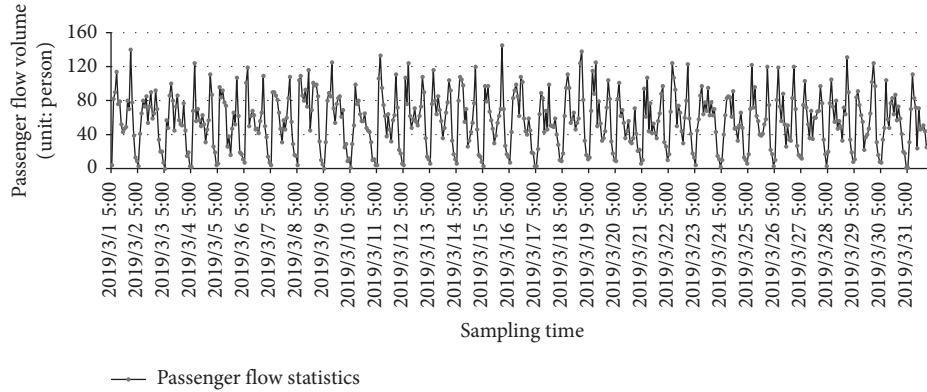


FIGURE 3: Passenger flow statistics within every 30 minutes of station 7 of line 28 in March.

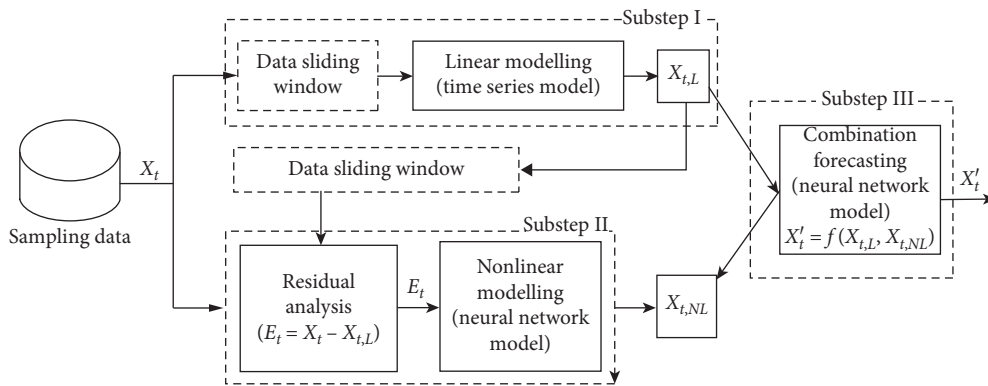


FIGURE 4: The time series data modelling in the first step.

changes based on the extracted feature of DBN. (III) Nonlinear combination analysis. In the substep, for the input data  $X_{t,L}$  and  $X_{t,NL}$ , the nonlinear model,  $X'_t = f(X_{t,L}, X_{t,NL})$ , is used to analyse and describe the relationship between the time series forecasting results and corresponding residual series, which aims to maximize the combination performance.

Figure 5 shows the process of second step; it receives several input data:  $X_t^A, \dots, X_t^K$ . These data could be from different spatial cross-correlative sections or spatial and temporal points. In real application,  $X_t^A, \dots, X_t^K$  may be simple linear relationship,  $X'_t = \alpha_A X_t^A + \dots + \alpha_K X_t^K$ , or also may be complex nonlinear relationship,  $X'_t = f(X_t^A, \dots, X_t^K)$ , so it does not employ only linear model or nonlinear model simply to describe the relationship among the input datasets. In the hybrid model, Im-ELM is employed to resolve the problem.

The “Forecasting Model A”, ..., “Forecasting Model K” are composite prediction models, which are described in Figure 4.

**3.1. Data Sliding Window.** The key point is to determine the data volume before each training or testing in the substep (I) and (II). Referring the idea of online sequential algorithms [41], the self-adaptive sliding data window is employed to represent the system dynamics [41–43], and it can also adjust

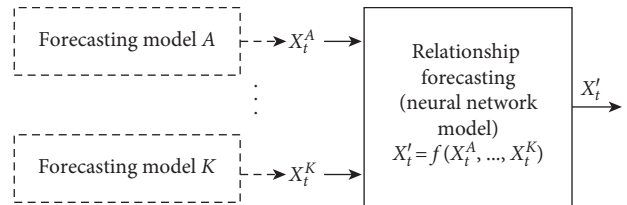


FIGURE 5: The relationship forecasting in the second step.

the structure of the time series model and the neural network model dynamically. The data sliding window is a first-in-first-out data sequence; its width can be fixed or dynamically adjusted.

In real application, the data are received one by one or chunk by chunk, and the sliding window is updated accordingly by adding new data and discarding the foremost ones; when the volume of received data  $s$  is less than the sliding window width  $N$ ,  $s \geq 1$  and  $s < N$ . However, in some extreme conditions, the received data volume  $s$  is larger than the sliding window width  $N$ ,  $s > N$ . For the continuity of the input data in the first substep (I) and the second substep (II), the foremost  $l$  data are selected from  $s$  receiving data, while  $l < s$  and  $l < N$ .

The data sliding window is expressed by the input samples and corresponding output results, which are the data pairs, as shown in the following equation:

$$W_{SD} = [(X_{t-N+1}, Y_{t-N+1}), \dots, (X_{t-1}, Y_{t-1}), (X_t, Y_t)], \quad (11)$$

where  $N$  is the width of the sliding window  $W_{SD}$  and also denotes the number of the pairs of input data and output result and  $t$  is the time index which shows the newest data index.

**3.2. Time Series Forecasting Model.** The time series forecasting is an important phase in the whole model; it is mainly used to analyse the real-time data and historical data and forecast the changes of the time series data in future. Generally, the time series analysis establishes the mathematical models by using curve fitting and parameter estimation. The basic model is ARMA, and its mathematical description is shown in the following equation:

$$Y_t = \sum_{i=1}^p \varphi_i Y_{t-i} + \sum_{j=0}^q \theta_j \varepsilon_{t-j}, \quad (12)$$

where  $Y_t$  are the time series data;  $\varphi$  is the coefficient of the autoregressive model;  $p$  is the order of the autoregressive model;  $\varepsilon$  is the white noise series, which fits normal distribution with the zero mean;  $\theta$  is the coefficient of the moving average model; and  $q$  is the order of moving average model.

ARMA ( $p, q$ ) is used to analyse the stationary stochastic process, but the time series data in some fields are changing upward and downward dramatically, and it also shows the characteristics of periodic fluctuation. The time series data are nonstationary stochastic process, which could be modelled by ARIMA ( $p, d, q$ ), shown as equation (13). It is the ARMA ( $p, q$ ) model with the differential operation.

$$\varphi_p(B)(1-B)^d Y_t = \theta_q(B)\varepsilon_t, \quad (13)$$

where  $B$  is the backward shift operator,  $BY_t = Y_{t-1}$ , and  $d$  is the order of differencing.

**3.3. Improved Extreme Learning Machine Model.** ELM is employed to forecast the data next time, which is a special learning algorithm for the single hidden layer feed-forward neural network. It only needs to determine the number and the output weight of the hidden layer neurons. The input weight and the threshold of activation function are set randomly and remain unchanged.

Given the training samples, the output of ELM is  $H\beta = Y$ , where  $H = [h_1, \dots, h_L]_{N \times L}$ ,  $h_{ij} = G(\omega_i, x_j, b_i)$ , is the output matrix of the hidden layer.  $\beta = [\beta_1, \dots, \beta_L]_{L \times m}^T$  is the output weight of the hidden layer connecting the hidden layer and output layer.  $Y = [y_1, \dots, y_N]_{N \times m}^T$  are the output results of ELM. After fixing the hidden layer neurons, ELM aims to find the optimal output weight matrix  $\beta$ , in order to make the output error least, so based on the theory of ELM, the optimal result is as follows:

$$\beta = H^\dagger Y = (H^T H)^{-1} H^T Y. \quad (14)$$

For improving self-adaptability, Huang et al. [44, 45] updated ELM and proposed an incremental extreme

learning machine algorithm (I-ELM). The basic idea of I-ELM is to update  $\beta$  dynamically by the residual  $e_L$  and output  $H_{L+1}$  before and after adding new hidden layer neuron, as shown in the following equation:

$$\beta = \frac{e_L, H_{L+1}}{\|H_{L+1}\|^2}. \quad (15)$$

For improving the stability and generalization ability of I-ELM, Im-ELM is proposed, in which the parameters  $\omega_{k,L+1}$  and  $b_{L+1}$  of the new adding neurons are not generated randomly. Referring the idea of literature [46, 47], the two parameters  $\omega_{k,L+1}$  and  $b_{L+1}$  are dynamically determined based on the chaos optimization algorithm (COA), which is highly efficient in global searching ability. In COA, the chaotic states are introduced into the optimization variables. The ergodic range of chaos is mapped into the range of optimization variables, and during the first and second carrier wave, shown as equations (16) and (17), to find the optimal solutions meeting the termination conditions.

$$X_{ij}^t = (b-a) \cdot d \cdot \gamma_{ij}^t, \quad (16)$$

$$X_{ij}^t = X_{ij}^{\text{best}} + \alpha(t) \cdot \gamma_{ij}^t, \quad (17)$$

where  $i = 1, \dots, n$  represents the optimization variables;  $j = 1, \dots, p$  represents the optimization variables mapped by the multiple chaos variables;  $[a, b]$  is the definition domain of  $X_{ij}$ ;  $t$  represents the iteration number; and  $d$  is the amplification gain. For better prediction accuracy, equation (17) is transformed to the following form:

$$X_{ij}^t = X_{ij}^{\text{best}} + \alpha(t) \cdot (\gamma_{ij}^t - \kappa), \quad j = 1, \dots, \frac{p}{2}, \quad (18)$$

$$X_{ij}^t = a + \gamma_{ij}^t \cdot (b-a), \quad j = \frac{p}{2} + 1, \dots, p,$$

where  $\kappa$  is a regulator. The optimization objective function,  $f(\cdot) = \sum_{i=1}^N (y_i - y'_i)^2 / N$ ,  $y_i$  and  $y'_i$  are target value and prediction value of  $i$ -th sample, respectively, and  $y'_i$  can be calculated by using equation (15).

Based on the analysis above, Im-ELM is described in detail as follows:

Im-ELM

Initialization:  $L = L_0$ ,  $\varepsilon = Y$

Training:

While  $L < L_{\max}$  &&  $E > \varepsilon$

S1: Increase a new hidden layer neuron,  $L = L + 1$

S2: Generate the input weight  $\omega_L$  and the bias  $b_L$  of the new neuron and optimize the parameters  $\omega_L$  and  $b_L$  based on the COA

S3: Calculate the new weight  $\beta_L$  based on equation (15) after increasing the new neurons

S4: Calculate the output error of the new neural network  $E_L = f(E_{L-1}, \beta_L H_L)$

**3.4. Deep Learning Model.** DBN is the generative one of the deep learning models, which is employed in the paper. It consists of the stacked restricted Boltzmann machine

(RBM) [48] with only one hidden layer [49]. Because of the integrating feature learning and deep learning, it has fast analysing and high data fitting ability [50, 51].

RBM is a special kind of Markov random field which consists of two parts: one is the visible layer and the other is the hidden layer. In each RBM, the visible variable  $v$  is to connect with the hidden units  $h$  by undirected weights  $\omega$  [52]. RBM is considered as an energy model; its energy is defined as [53]

$$E(v, h) = -\alpha^T v - b^T h - h^T \omega v. \quad (19)$$

The DBN stacks several RBMs as the unsupervised network considering the visible layer to the hidden layer. The hidden layer of one RBM is the visible layer of the subsequent RBM, and the training process can be divided into two steps. One step is the unsupervised learning. In this process, the training samples are transformed through layer by layer, and the better initial parameters  $\omega_{i,j}$ ,  $\alpha_i$ , and  $b_i$  could be obtained.  $\omega_{i,j}$  is the symmetric weight connecting the hidden unit  $j$  and the visible unit  $i$ .  $\alpha_i$  is the bias of the visible unit, and  $b_i$  is the bias of the hidden unit. The other step is the supervised learning. In this step, some learning algorithms are used to optimize the parameters obtained in the first step. At last, through global fine-tuning process, the optimal parameters are selected. After finishing the training of the RBMs, the DBN features can be extracted from the topmost hidden layer [54].

For determining the parameters  $\alpha$ ,  $b$ , and  $\omega$  of each RBM, in the first step of training, the contrast divergence (CD) algorithm is adopted to train each RBM one by one [48, 55]. CD algorithm is a fast learning algorithm, with one-step Gibbs sampling for making a better approximation. The process of CD is mainly in four steps: (1) set the first visible layer variables as the input samples; (2) from visible layer to hidden layer, the hidden layer variables are updated by  $P(h_j = 1 | v)$  based on known visible layer states; (3) negative phase: based on the hidden layer states in the second step, the visible layer is reconstructed by  $P(v_i = 1 | h)$ ; (4) update the weights. The updating criterion of parameters is as follows:

$$\begin{aligned} \Delta \omega_{i,j} &= \eta (\langle v_i h_j \rangle - \langle v_i' h_j' \rangle), \\ \Delta \alpha_i &= \eta (\langle v_i \rangle - \langle v_i' \rangle), \\ \Delta b_j &= \eta (\langle h_j \rangle - \langle h_j' \rangle), \end{aligned} \quad (20)$$

where  $v_i$  and  $v_i'$  are the states of  $i$ -th neuron before and after reconstructing visible layer respectively;  $h_j$  and  $h_j'$  are the states of  $j$ -th neuron before and after reconstructing visible layer; and  $\eta$  is the learning rate.

## 4. Experiments and Analysis

For illustrating and verifying the model proposed in the paper, passenger flow statistics at station 7 of line 28 and station 8 of line 10 are used as experimental samples. The sampling period of the original dataset is from October 1, 2018, to March 31, 2019, for a total of 6 months. The length of time is from 7:30 to 9:30 in the morning peak hours and

from 16:30 to 18:30 in the evening peak hours. As the model input, the original dataset is aggregated into a time series of 30-minute interval based on equations (6)–(10), and the data from station 7 of line 28 are shown in Figure 6. Moreover, the performances of models are evaluated by the mean-squared error (MSE) and mean absolute percent error (MAPE).

$$\begin{aligned} \text{MSE} &= \frac{1}{N} \sum_{i=1}^N (y_i - y_i')^2, \\ \text{MAPE} &= \frac{1}{N} \sum_{i=1}^N \frac{\|y - y_i'\|}{\|y_i'\|}. \end{aligned} \quad (21)$$

**4.1. Time Series Model.** The following presents the modelling process for passenger flow analysis using time series model, which is based on the content described in the literature [5].

**4.1.1. Unit Root Test.** By observing Figure 6, the sequence shows significant instability. After the first-order difference, the unit root test results are shown in Table 2. When the additional item is “Intercept,” supposing  $H_0: \delta = 0$ , the t-Static of unit root is  $-7.962247$ , and it is obviously less than the 1%, 5%, and 10% significance level. The critical values of t-static are  $-3.886751$ ,  $-3.052169$ , and  $-2.666593$  separately. Obviously, the statistical value of  $\zeta$  test is less than the corresponding DW critical values. As a result, it means that after the first-order difference analysis, the data sequence is stable and could be analyzed by time series models.

**4.1.2. Model Selection.** The process of selecting the time series model is identifying the orders of the autoregressive ( $p$ ) and moving average terms ( $q$ ). The orders can be obtained by calculating the autocorrelation functions (ACFs) and partial autocorrelation functions (PACFs) of the sequence, and the general judgment rules are shown in Table 3 [56]. Analysing these results in Figure 7, when lag = 2, both ACF and PACF show the tails off exponentially. According to Table 3, the ARMA model is selected preliminarily, and its parameters  $p$  and  $q$  are limited in intervals [1, 2].

**4.1.3. Parameters Estimation.** The fitting degree needs to be tested based on the information criteria AIC, SC, and HQC, so as to determine the lag order  $p$  and  $q$  and other parameters. Item included in test equation. Based on their interval, the four models ARIMA (2, 1, 2), ARIMA (2, 1, 1), ARIMA (1, 1, 2), and ARIMA (1, 1, 1) are constructed. After setting the sample size 40, 50, and 60, each model is tested three times. Finally, the minimum of AIC, SC, and HQC can be obtained, as shown in Table 4. Analysing the results, the minimum of AIC is 7.108482, SC is 7.306342, and HQC is 7.135764. After comprehensive analysis, the performances of ARIMA (2, 1, 2) in the four models are best, which is selected as the time series model.

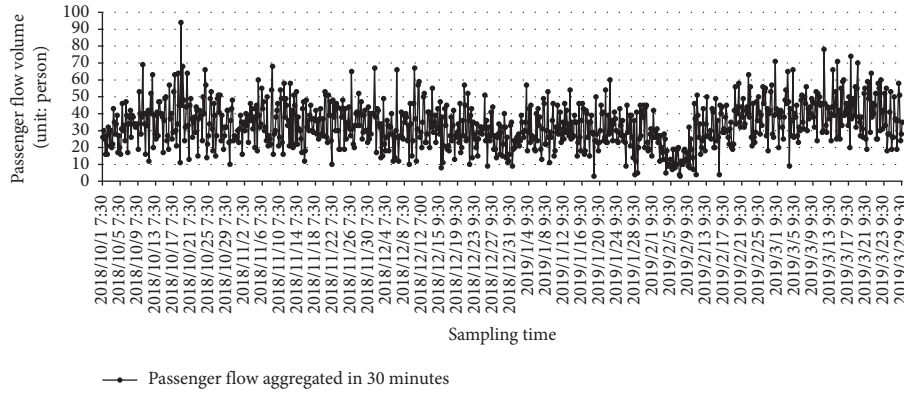


FIGURE 6: Part of the input dataset at station 7 of line 28.

TABLE 2: The results of unit root tests.

Item included in test equation	ADF test statistic	t-Static		
		1% level	5% level	10% level
None	-8.059761	-2.708094	-1.962813	-1.606129
Intercept	-7.962247	-3.886751	-3.052169	-2.666593
Trend and intercept	-7.729247	-4.616209	-3.710482	-3.297799

TABLE 3: ACF and PACF of the time series model.

	AR ( $p$ )	MA ( $q$ )	ARMA ( $p, q$ )
ACF	Tails off exponentially	Cuts off after lag $q$	Tails off exponentially
PACF	Cuts off after lag $p$	Tails off exponentially	Tails off exponentially

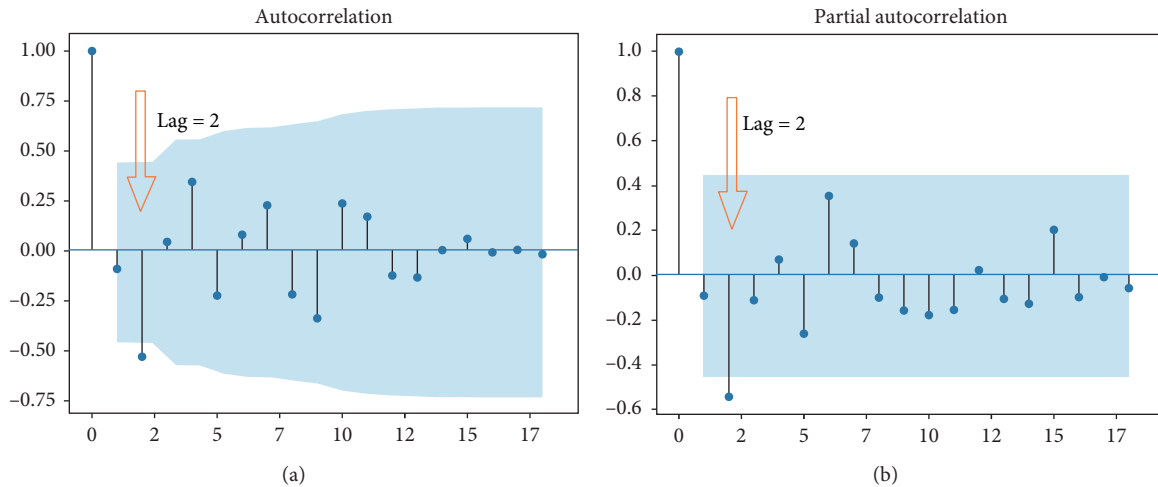


FIGURE 7: The correlation analysis.

4.1.4. *Model Testing.* For verifying the performances of the selected model, we need to test whether the residual series is the white noise sequence or not by calculating the ACF and PACF. The results are shown in Figures 8 and 9. In Figure 8, the lag order shows the residual series is a white noise sequence obviously. Figure 9 shows the fitting curve of the real-world data and forecasting results of the early peak

period. The fitting effect is significantly reduced due to traffic jams in the morning peak. The conclusion proves that the single linear model cannot well describe the nonlinear factors affecting passenger flow change.

In the paper, historical data in a large time scale are selected as the assistant to improve the predicting effects, and the analysing model of historical data need to be



TABLE 4: AIC, SC, and HQC of different ARIMA models.

Model	AIC	SC	HQC
ARIMA (2, 1, 2)	7.108482	7.306342	7.135764
ARIMA (2, 1, 1)	7.198449	7.346844	7.218911
ARIMA (1, 1, 2)	7.238309	7.387430	7.263546
ARIMA (1, 1, 1)	7.236094	7.335508	7.252919

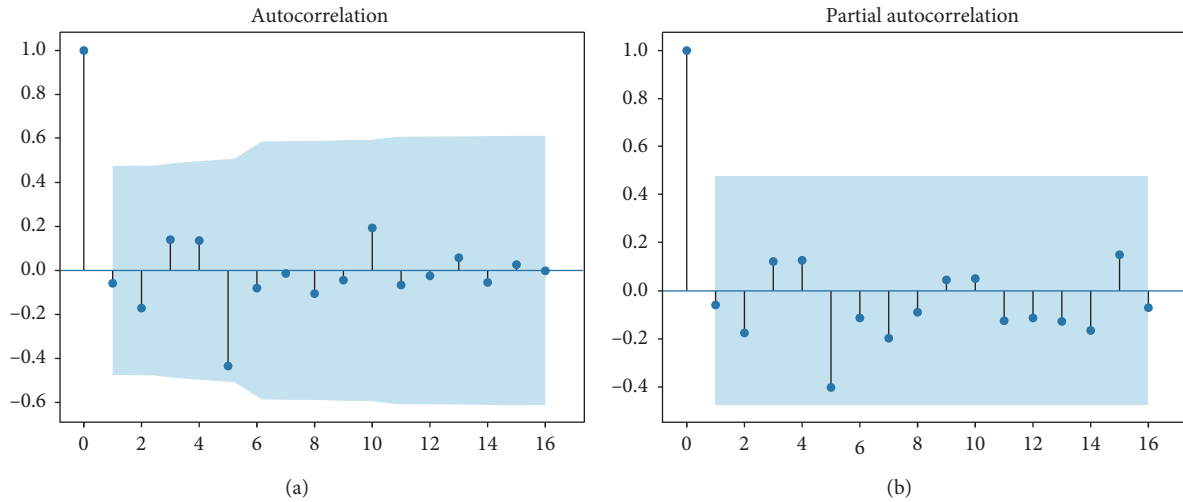


FIGURE 8: The correlation analysis of ARIMA (2, 1, 2).

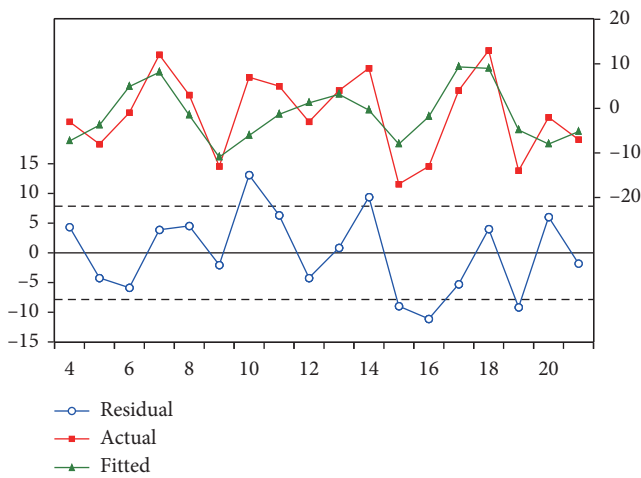


FIGURE 9: The fitting analysis of ARIMA (2, 1, 2).

determined based on the operations above. After analysing, ARIMA (2, 1, 2) is selected for the historical data.

4.2. *Im-ELM Model Analysis.* Figures 10 and 11 show the changes of the number of hidden layer neurons and the learning error during the training process, and  $X^{t+1} = \sin(2/X^t)$  is selected as the chaotic map function, and regulator  $\kappa = 0.45$ . As the number of hidden neurons increases, the training error is decreasing. From Figure 11, the error decreases rapidly at first, and when the number of

hidden neurons is more than 20, the error tends to be stable. The final result is acceptable.

For testing the performances of Im-ELM, a piece of passenger time series data is selected from the whole data. Figure 12 shows the training and validating process. With the increasing training process, the training error decreases, and the error is stable and reaches optimal results when MSE is around 10. The forecasting results of the real-world data are also around 10. Table 5 shows the comparison between Im-ELM and other models (SA-ELM [57], ImSAP-ELM [58], and ELM). Because of introducing COA, more time is needed to optimize the parameters of the new neurons in each iteration. The training time is more than SA-ELM, but less than ImSAP-ELM and ELM. The difference is only about 0.12 seconds between Im-ELM and SA-ELM, which is totally acceptable after comprehensive analysis. The number of hidden layer neurons of Im-ELM is 23 less than the others, and the accuracy of Im-ELM is the best, which is suitable in the hybrid model and applications.

4.3. *DBN Analysis.* In DBN, the number of hidden layers and hidden neurons is determined by the enumeration method layer by layer. Table 6 shows the comparison between DBN and other models. The prediction accuracy of DBN increases with number of hidden layers. However, too many hidden layers or neurons may reduce prediction accuracy. From Table 6, the DBN-4 (with 3 hidden layers and 150 neurons in each hidden layer) performs best, and selected as the part of HTSDBNE.

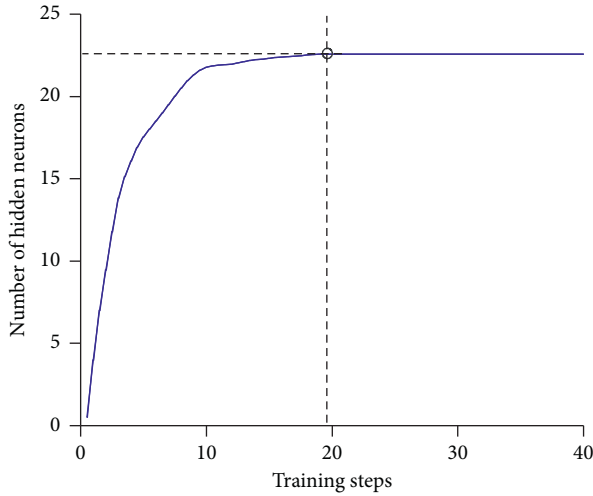


FIGURE 10: The variation curve of the number of hidden neurons.

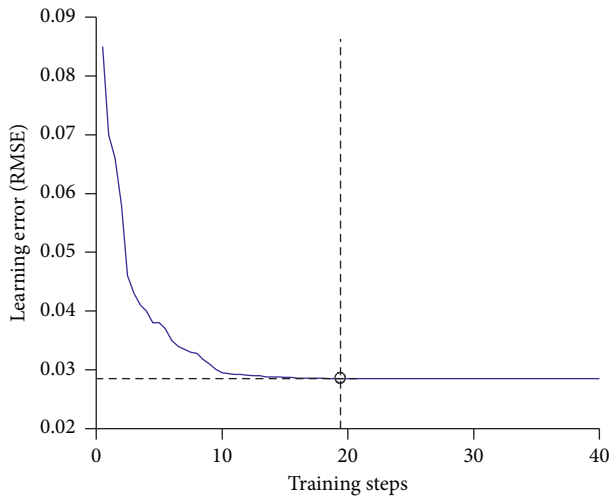


FIGURE 11: The variation curve of training error.

**4.4. Hybrid Model Testing and Analysis.** The time series data of the working day and nonworking day are obtained to test the proposed model HTSDBNE. Tables 7 and 8 and Figures 13 and 14 show the comparison between the HTSDBNE and other models (ARIMA (2, 1, 2), ELM, TS-ANN, SLMBP, SAE-DNN [59], and MPDF [21]). In the passenger off-peak time, such as 7:30 am, all models show good accuracy and small MSE and MAPE. However, in the peak time of the working day and nonworking day, especially in the critical zone, the performance of HTSDBNE shows greater superiority. For example, in Figure 13, at 8:30 am point, the MSE of HTSDBNE is 8.24, far lower than the error 18.13 of ARIMA (2, 1, 2), and compared with others, the accuracy is improved significantly. In few time points (marked in bold in Tables 7 and 8 and red asterisk in Figures 13 and 14), the result of HTSDBNE is weaker than SAE-DNN and MPDF, but the difference is very small, and the largest difference is only 1.4257%.

Figures 15 and 16 show the changing trends of the real-world data and the forecasting results in each working day of

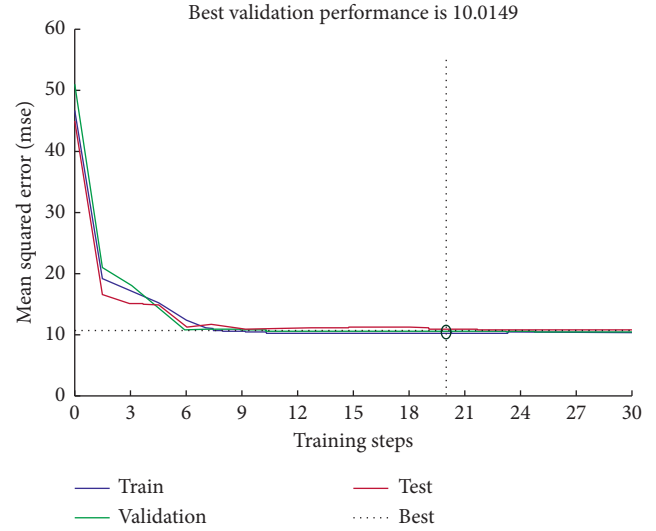


FIGURE 12: The training and validation of Im-ELM.

TABLE 5: The comparison between Im-ELM and other models.

	Im-ELM	SA-ELM	ImSAP-ELM	ELM
Number of hidden neurons	23	29	30	57
Training time (s)	2.7518	2.6287	3.0853	3.9837
Error (MSE)	10.015	11.254	11.438	13.583

TABLE 6: Comparison between DBN and other models.

Models	Hidden layers	Hidden neurons	MSE	MAPE (%)
DBN-1	2	200-200	14.524	15.968
DBN-2	2	150-150	14.305	15.208
DBN-3	3	200-200-200	13.624	14.483
<b>DBN-4</b>	<b>3</b>	<b>150-150-150</b>	<b>12.385</b>	<b>13.167</b>
DBN-5	4	150-150-150-150	13.747	14.615
LSSVM	—	50	12.472	13.259
ELM	—	60	14.983	15.115

the station 7 of line 28. Due to space constraints and the similarity of the results, the analysis of station 8 of line 10 is omitted here. From the curve fitting, HTSDBNE is better in time 9:00 to 9:30 and 16:30 to 18:30. The difference between real-world data and forecasting results of HTSDBNE is around 2.5, and the least difference of other models is around 3.2. The HTSDBNE has the best performances compared with other models. At 7:30 am, ARIMA (2, 1, 2) does not capture the real changes of the passenger flow, and at 8:00 and 17:00 pm, the road section is in the peak and congestion; these led to a large forecasting delay for ELM, TS-ANN, and SLMBP. HTSDBNE, SAE-DNN, and MPDF are relatively successful in capturing the sharp changes of the passenger flow, but HTSDBNE is better in the forecasting accuracy. In the end of the early peak and evening peak hours, the prediction results of HTSDBNE are most consistent with the actual situation. The results indicate that HTSDBNE shows better performances and applicability than other models in both peak and off-peak time.

TABLE 7: Comparison between HTSDBNE and other models (MAPE) (station 7 of line 28).

Models	ARIMA (2, 1, 2) (%)	ELM (%)	TS-ANN (%)	SLMBP (%)	SAE-DNN (%)	MPDF (%)	HTSDBNE (%)	
Working day	7:30	24.59	23.58	19.91	19.57	15.00	14.80	11.84
	8:00	25.49	24.66	20.82	19.64	15.38	15.17	13.91
	8:30	23.82	22.89	19.33	18.87	14.52	14.33	11.46
	9:00	24.64	23.59	19.92	19.58	15.01	14.82	12.12
	9:30	23.66	22.72	19.18	18.85	<b>11.25</b>	12.01	11.41
	16:30	21.92	21.52	18.17	18.68	12.59	13.18	12.06
	17:00	22.97	22.15	18.70	18.38	14.09	13.91	11.13
	17:30	23.88	21.97	19.13	18.49	14.30	14.11	13.29
	18:00	23.58	22.54	19.04	18.71	14.34	14.15	11.32
	18:30	22.39	20.60	17.97	17.36	13.42	13.25	11.68
Nonworking day	7:30	21.71	20.88	17.63	17.33	13.29	13.11	10.49
	8:00	20.69	18.59	16.28	15.69	12.15	11.99	9.88
	8:30	23.19	20.18	18.19	17.27	13.47	13.30	10.64
	9:00	17.09	16.58	14.00	13.76	10.55	<b>10.41</b>	10.43
	9:30	15.37	14.98	12.74	12.47	9.58	9.45	8.56
	16:30	15.29	14.63	12.36	12.14	10.12	9.39	8.35
	17:00	15.38	13.97	11.92	11.65	8.96	8.84	8.39
	17:30	14.59	14.12	12.39	11.93	8.52	<b>8.12</b>	8.40
	18:00	16.33	14.81	12.83	12.44	8.35	8.84	8.58
	18:30	14.39	13.59	11.47	11.28	8.65	8.53	8.03

TABLE 8: Comparison between HTSDBNE and other models (MAPE) (station 8 of line 10).

Models	ARIMA (2, 1, 2) (%)	ELM (%)	TS-ANN (%)	SLMBP (%)	SAE-DNN (%)	MPDF (%)	HTSDBNE (%)	
Working day	7:30	24.59	23.58	19.91	19.57	15.00	14.80	11.84
	8:00	25.49	24.66	20.82	19.64	15.38	15.17	13.91
	8:30	23.82	22.89	19.33	18.87	14.52	14.33	11.46
	9:00	24.64	23.59	19.92	19.58	15.01	14.82	12.12
	9:30	23.66	22.72	19.18	18.85	<b>11.25</b>	12.01	11.41
	16:30	21.92	21.52	18.17	18.68	12.59	13.18	12.06
	17:00	22.97	22.15	18.70	18.38	14.09	13.91	11.13
	17:30	23.88	21.97	19.13	18.49	14.30	14.11	13.29
	18:00	23.58	22.54	19.04	18.71	14.34	14.15	11.32
	18:30	22.39	20.60	17.97	17.36	13.42	13.25	11.68
Nonworking day	7:30	21.71	20.88	17.63	17.33	13.29	13.11	10.49
	8:00	20.69	18.59	16.28	15.69	12.15	11.99	9.88
	8:30	23.19	20.18	18.19	17.27	13.47	13.30	10.64
	9:00	17.09	16.58	14.00	13.76	10.55	<b>10.41</b>	10.43
	9:30	15.37	14.98	12.74	12.47	9.58	9.45	8.56
	16:30	15.29	14.63	12.36	12.14	10.12	9.39	8.35
	17:00	15.38	13.97	11.92	11.65	8.96	8.84	8.39
	17:30	14.59	14.12	12.39	11.93	8.52	<b>8.12</b>	8.40
	18:00	16.33	14.81	12.83	12.44	8.35	8.84	8.58
	18:30	14.39	13.59	11.47	11.28	8.65	8.53	8.03

Figures 17 and 18 show the changing trends of real-world data and forecasting data in each nonworking day of station 7 of line 28. The passenger flow peak hour of nonworking days is 30 minutes later than the working days. In Figure 17, the passenger flow is on the rise sharply from 8:00 am to 8:30 am. ARIMA (2, 1, 2), ELM, TS-ANN, SLMBP, SAE-DNN, and MPDF capture this upward trend, but the accuracy of prediction results is not high, and the maximum error is more than 20%. However, HTSDBNE performs well with an error of only 8.787%. In the end of the morning peak, HTSDBNE has successfully forecasted the real status of the passenger flow. Different from the morning peak of nonworking days, in the evening peak hours shown in Figure 18, the passenger flow increases gradually, and the three hybrid

models (SAE-DNN, MPDF, and HTSDBNE) describe this characteristic. The HTSDBNE is the best in the forecasting accuracy, and the maximum difference is only about 2.

## 5. Conclusions

In this paper, the original passenger flow statistical data were deeply analysed and constructed as a time series with an aggregation interval of 30 minutes. Based on the characteristics of passenger flow variation, a novel hybrid forecasting model, HTSDBNE, is proposed, which consisted of ARIMA, DBN, and Im-ELM. In the first step, the ARIMA is used to analyse the stability of time series sequences of the historical data and real-time data and then make a

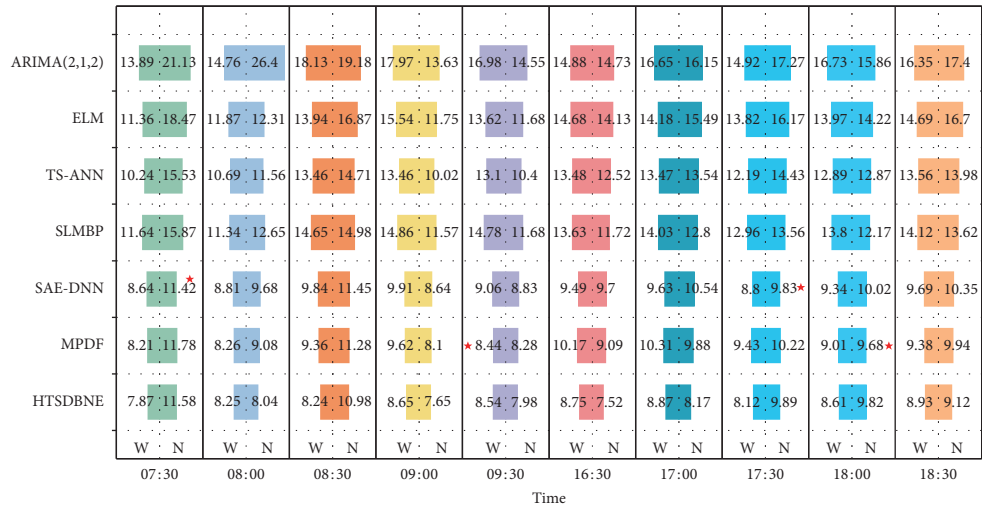


FIGURE 13: Comparison between HTSDBNE and other models (MSE) (station 7 of line 28). The column W shows the results of working day, and column N shows the results of nonworking day.

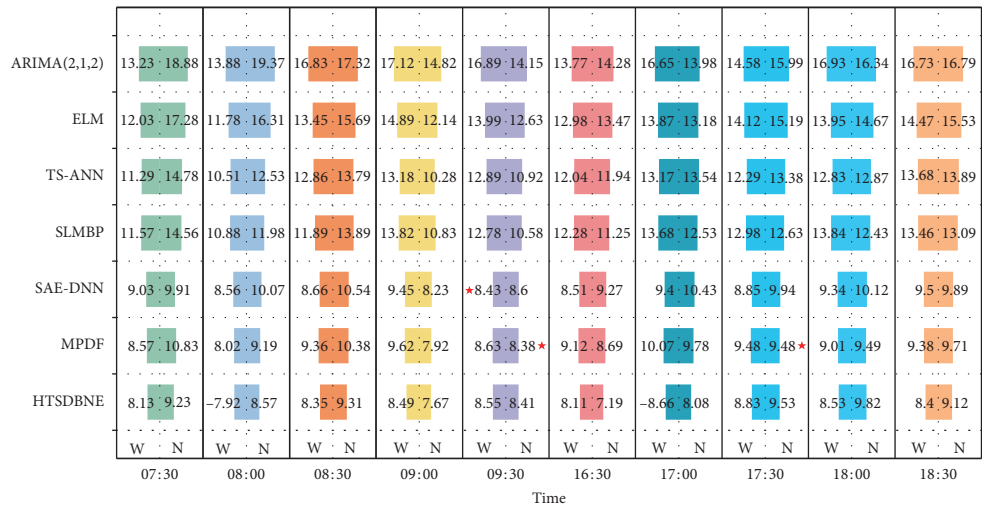


FIGURE 14: Comparison between HTSDBNE and other models (MSE) (station 8 of line 10). The column W shows the results of working day, and column N shows the results of nonworking day.

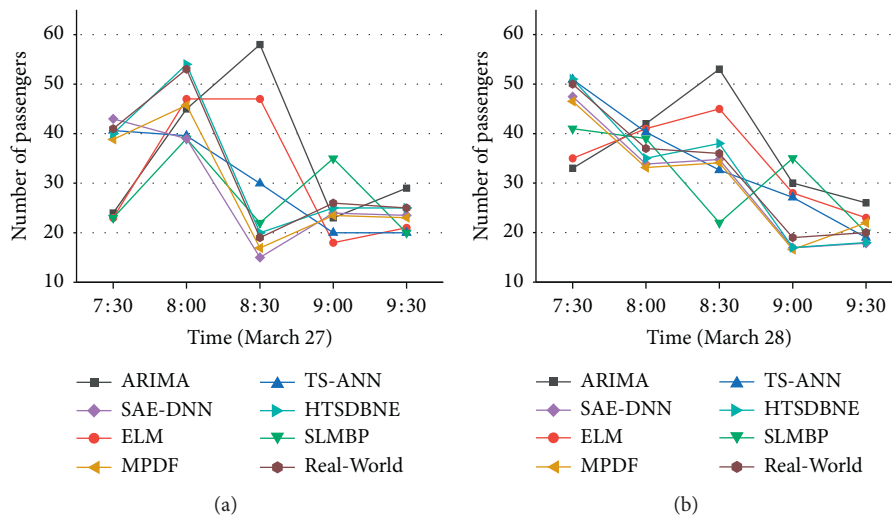


FIGURE 15: Continued.

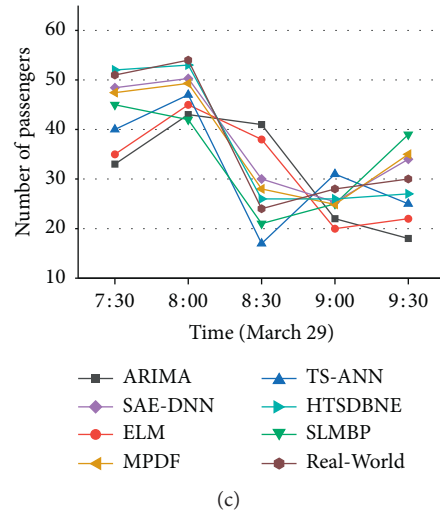


FIGURE 15: Comparison between HTSDBNE and other models in the morning peak hours of working days (station 7 of line 28).

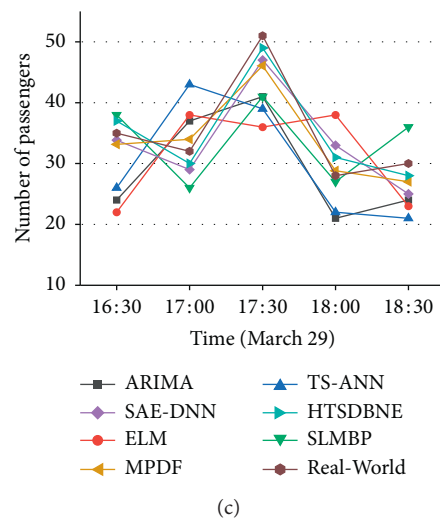
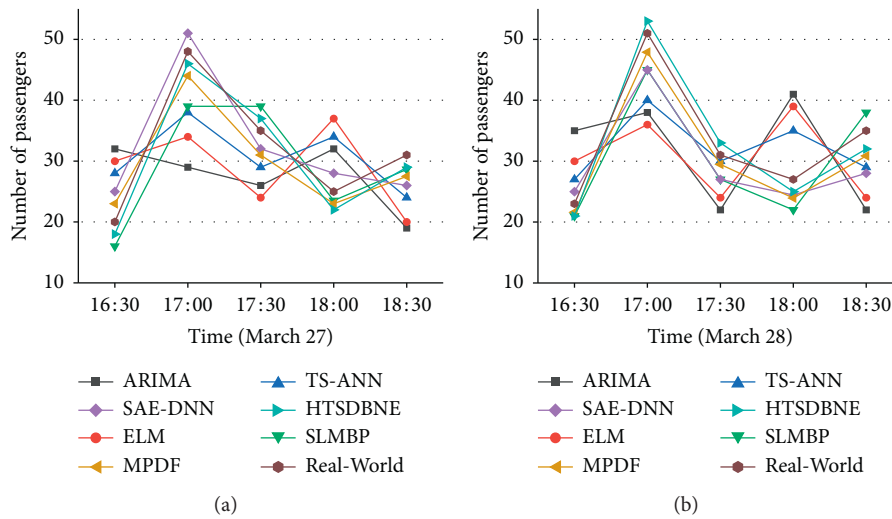


FIGURE 16: Comparison between HTSDBNE and other models in the evening peak hours of working days (station 7 of line 28).

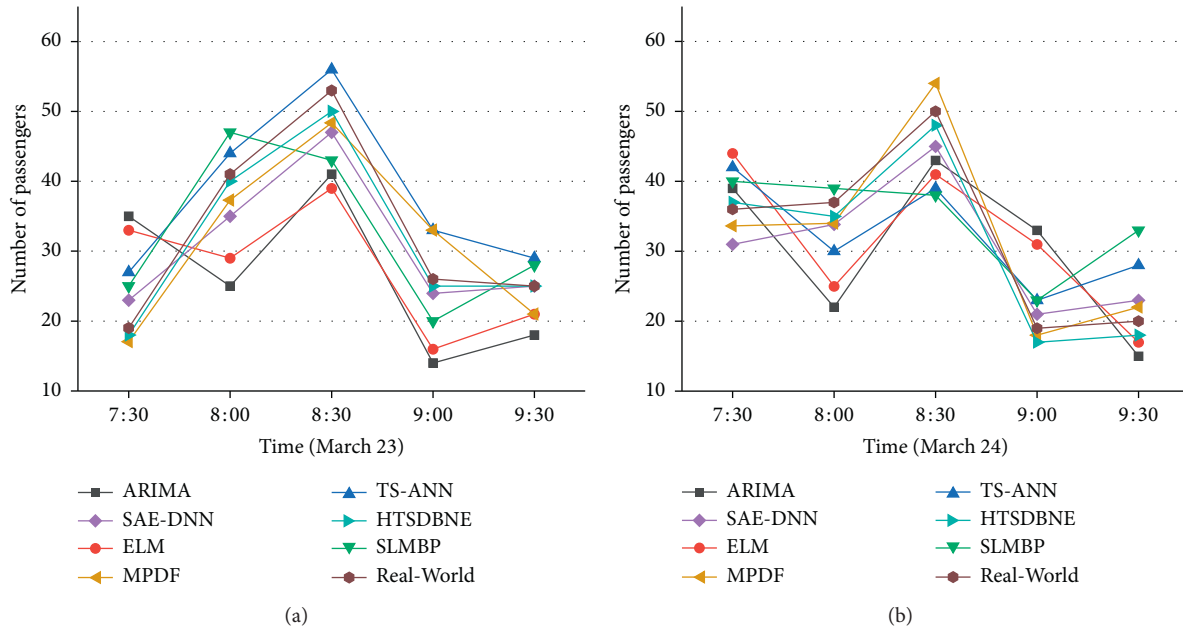


FIGURE 17: Comparison between HTSDBNE and other models in the morning peak hours of nonworking days (station 7 of line 28).

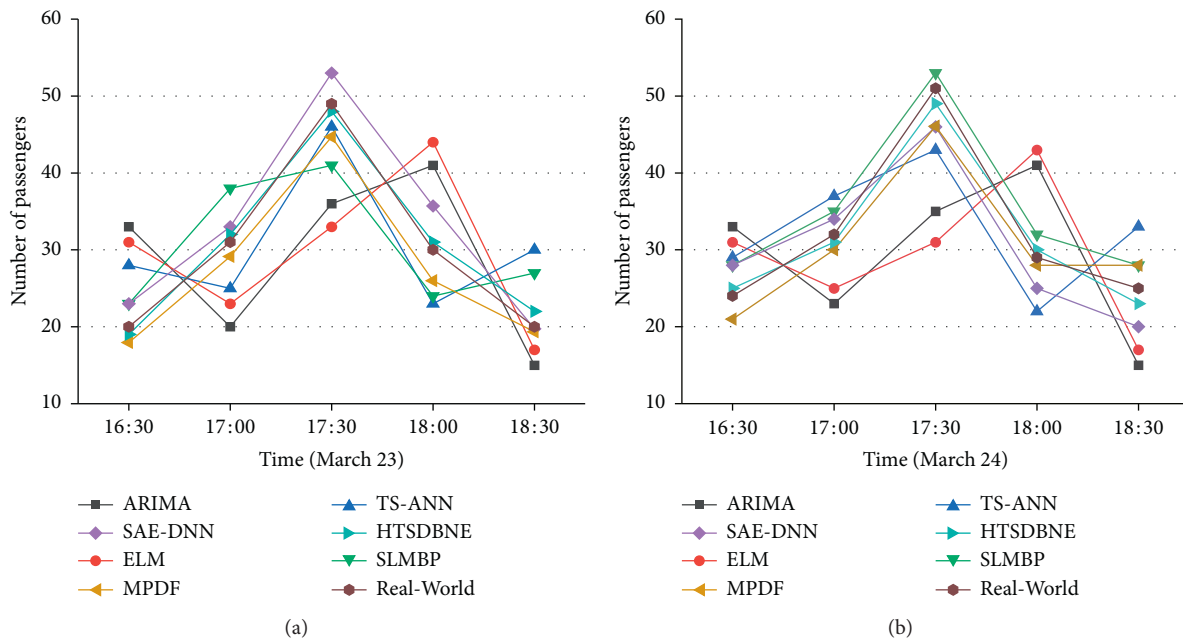


FIGURE 18: Comparison between HTSDBNE and other models in the evening peak hours of nonworking days (station 7 of line 28).

preliminary prediction. Next, a subhybrid model based on DBN and Im-ELM is constructed, which is used to extract the features of the residual series of the forecasting results and forecast the changes of the residual series. At the end of first step, the residual and preliminary forecasting results are reassembled together as the intermediate prediction results. For making effective use of historical data to improve the current forecasting, in the first step of HTSDBNE, the parallelized submodels are employed, which have the same structure as the one above. In the second step, the

intermediate forecasting results from the first step based on the real-time passenger flow and historical data in different time scales are reassembled together by Im-ELM as the final forecasting results. In order to verify the performances of HTSDBNE, a comparative analysis is performed with ARIMA(2, 1, 2), ELM, TS-ANN, SLMBP, SAE-DNN, and MPDF. The testing results indicate that the novel hybrid model can capture and forecast the changes of passenger flow accurately in a more general environment. In future research, the hybrid model is considered to be parallelized in

the big data computing environment, in order to improve the computational efficiency to adapt the real-time forecasting of all bus lines in the whole city.

## Data Availability

The bus passenger flow data used to support the results of this study are included within the article.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was partially supported by the Fundamental Research Funds for the Central Universities (grant nos. 3132019205 and 3132019354), Liaoning Provincial Natural Science Foundation of China (grant no. 20170520196), and Scientific Research Fund of Liaoning Provincial Education Department (grant nos. JYT2019LQ01 and JYT2019LQ02).

## References

- [1] B. L. Smith, B. M. Williams, and R. Keith Oswald, "Comparison of parametric and nonparametric models for traffic flow forecasting," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 303–321, 2002.
- [2] D. Sun, S. Chen, C. Zhang, and S. Shen, "A bus route evaluation model based on GIS and super-efficient data envelopment analysis," *Transportation Planning and Technology*, vol. 39, no. 4, pp. 407–423, 2016.
- [3] L. Qin, W. Li, and S. Li, "Effective passenger flow forecasting using STL and ESN based on two improvement strategies," *Neurocomputing*, vol. 356, pp. 244–256, 2019.
- [4] G. Nian, F. Chen, Z. Li, Y. Zhu, and D. Sun, "Evaluating the alignment of new metro line considering network vulnerability with passenger ridership," *Transportmetrica A: Transport Science*, vol. 15, no. 2, pp. 1402–1418, 2019.
- [5] H. Zhai, L. Cui, Y. Nie, X. Xu, and W. Zhang, "A comprehensive comparative analysis of the basic theory of the short term bus passenger flow prediction," *Symmetry*, vol. 10, no. 9, p. 369, 2018.
- [6] Z. Ma, J. Xing, M. Mesbah, and L. Ferreira, "Predicting short-term bus passenger demand using a pattern hybrid approach," *Transportation Research Part C: Emerging Technologies*, vol. 39, pp. 148–163, 2014.
- [7] R. Xue, D. J. Sun, and S. Chen, "Short-term bus passenger demand prediction based on time series model and interactive multiple model approach," *Discrete Dynamics in Nature and Society*, vol. 2015, Article ID 682390, 11 pages, 2015.
- [8] X. Yang and L. Liu, "Short-term passenger flow forecasting on bus station based on affinity propagation and support vector machine," *Journal of Wuhan University of Technology (Transportation Science and Engineering)*, vol. 40, no. 1, pp. 36–40, 2016.
- [9] S. Guo, W. Li, W. Bai, and D. Zhang, "Prediction of short-term passenger flow on a bus stop based on LSSVM," *Journal of Wuhan University of Technology (Transportation Science and Engineering)*, vol. 37, no. 3, pp. 603–607, 2013.
- [10] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: a survey," *Fuzzy Sets and Systems*, vol. 66, no. 1, pp. 1–13, 1994.
- [11] T.-H. Tsai, C.-K. Lee, and C.-H. Wei, "Neural network based temporal feature models for short-term railway passenger demand forecasting," *Expert Systems with Applications*, vol. 36, no. 2, pp. 3728–3736, 2009.
- [12] J. Roos, G. Gavin, and S. Bonnevey, "A dynamic bayesian network approach to forecast short-term urban rail passenger flows with incomplete data," *Transportation Research Procedia*, vol. 26, pp. 53–61, 2017.
- [13] S. Sun, C. Zhang, and G. Yu, "A bayesian network approach to traffic flow forecasting," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 1, pp. 124–132, 2006.
- [14] Lehman Center for Transportation Research, *Florida Advanced Public Transit Systems Program*, Lehman Center for Transportation Research, Miami, FL, USA, 2009.
- [15] C. Bishop, "Improving the generalization properties of radial basis function neural networks," *Neural Computation*, vol. 3, no. 4, pp. 579–588, 1991.
- [16] H. Li, Y. Wang, X. Xu, L. Qin, and H. Zhang, "Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network," *Applied Soft Computing*, vol. 83, Article ID 105620, 2019.
- [17] Y. Li, X. Wang, S. Sun, X. Ma, and G. Lu, "Forecasting short-term subway passenger flow under special events scenarios using multiscale radial basis function networks," *Transportation Research Part C: Emerging Technologies*, vol. 77, pp. 306–328, 2017.
- [18] H. Ling and E. Xi, "Prediction model of public transport passenger volume based on random fluctuations," *Journal of Chang'an University Natural Science Edition*, vol. 32, no. 1, pp. 85–88, 2012.
- [19] Q. Yang, Y. Yang, Z. Feng, and X. Zhao, "Prediction method of passenger volume of city public transit based on grey theory and markov model," *China Journal of Highway and Transport*, vol. 26, no. 6, pp. 169–175, 2013.
- [20] Q. Wang and Q. Zhang, "Forecasting of short-term urban public transit volume based on random gray ant colony neural network," *Application Research of Computers*, vol. 29, no. 6, pp. 2078–2080, 2012.
- [21] Y. Bai, Z. Sun, B. Zeng, J. Deng, and C. Li, "A multi-pattern deep fusion model for short-term bus passenger flow forecasting," *Applied Soft Computing*, vol. 58, pp. 669–680, 2017.
- [22] Y. Sun, B. Leng, and W. Guan, "A novel wavelet-SVM short-time passenger flow prediction in Beijing subway system," *Neurocomputing*, vol. 166, pp. 109–121, 2015.
- [23] J. Liu, Y. Heng, H. Zhao, X. Gao, and P. Wang, "A prediction model of short-term passenger flow for urban transit hubs," *Journal of Transport Information and Safety*, vol. 32, no. 2, pp. 41–44, 2014.
- [24] C. Wang, Z. Cao, X. Zhang, W. Zhang, and H. Zhai, "Study on the prediction model of short-term bus passenger flow based on big data," *MATEC Web of Conferences*, vol. 232, Article ID 02050, 2018.
- [25] J. C. F. de Winter, S. D. Gosling, and J. Potter, "Comparing the pearson and spearman correlation coefficients across distributions and sample sizes: a tutorial using simulations and empirical data," *Psychological Methods*, vol. 21, no. 3, pp. 273–290, 2016.
- [26] E. Vlahogianni and M. Karlaftis, "Temporal aggregation in traffic data: implications for statistical characteristics and model choice," *Transportation Letters*, vol. 3, no. 1, pp. 37–49, 2011.
- [27] Y. Liu, Z. Liu, and R. Jia, "DeepPF: a deep learning based architecture for metro passenger flow prediction,"

- Transportation Research Part C: Emerging Technologies*, vol. 101, pp. 18–34, 2019.
- [28] J. Ke, H. Zheng, H. Yang, and X. Chen, “Short-term forecasting of passenger demand under on-demand ride services: a spatio-temporal deep learning approach,” *Transportation Research Part C: Emerging Technologies*, vol. 85, pp. 591–608, 2017.
- [29] G. Liu, Z. Yin, Y. Jia, and Y. Xie, “Passenger flow estimation based on convolutional neural network in public transportation system,” *Knowledge-Based Systems*, vol. 123, pp. 102–115, 2017.
- [30] W. E. Allen and R. D. Murray, *Implementation of Smart Card Automatic Fare Collection (AFC) Technology in Small Transit Agencies for Standards Development*, Transportation Research Board, Washington, DC, USA, 2016.
- [31] S. Panigrahi and H. S. Behera, “A hybrid ETS-ANN model for time series forecasting,” *Engineering Applications of Artificial Intelligence*, vol. 66, pp. 49–59, 2017.
- [32] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003.
- [33] Y. Boureau and Y. L. Cun, “Sparse feature learning for deep belief networks,” in *Proceedings of the Advances in Neural Information Processing Systems*, pp. 1185–1192, Vancouver, Canada, December 2008.
- [34] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, “Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations,” in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 609–616, Montreal, Canada, June 2009.
- [35] S. Ding, N. Zhang, J. Zhang, X. Xu, and Z. Shi, “Unsupervised extreme learning machine with representational features,” *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 2, pp. 587–595, 2017.
- [36] S. Ding, H. Zhao, Y. Zhang, X. Xu, and R. Nie, “Extreme learning machine: algorithm, theory and applications,” *Artificial Intelligence Review*, vol. 44, no. 1, pp. 103–115, 2015.
- [37] M. Khashei and M. Bijari, “A novel hybridization of artificial neural networks and ARIMA models for time series forecasting,” *Applied Soft Computing*, vol. 11, no. 2, pp. 2664–2675, 2011.
- [38] J. J. Ruiz-Aguilar, I. J. Turias, and M. J. Jiménez-Come, “Hybrid approaches based on SARIMA and artificial neural networks for inspection time series forecasting,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 67, pp. 1–13, 2014.
- [39] P. R. A. Firmino, P. S. G. de Mattos Neto, and T. A. E. Ferreira, “Error modeling approach to improve time series forecasters,” *Neurocomputing*, vol. 153, pp. 242–254, 2015.
- [40] P. S. G. de Mattos Neto, G. D. C. Cavalcanti, and F. Madeiro, “Nonlinear combination method of forecasters applied to PM time series,” *Pattern Recognition Letters*, vol. 95, pp. 65–72, 2017.
- [41] J.-C. Yin, “A variable-structure online sequential extreme learning machine for time-varying system prediction,” *Neurocomputing*, vol. 261, pp. 115–125, 2017.
- [42] C.-Y. Chen, T.-H. S. Li, Y.-C. Yeh, and C.-C. Chang, “Design and implementation of an adaptive sliding-mode dynamic controller for wheeled mobile robots,” *Mechatronics*, vol. 19, no. 2, pp. 156–166, 2009.
- [43] C.-Y. Chen, T.-H. S. Li, and Y.-C. Yeh, “EP-based kinematic control and adaptive fuzzy sliding-mode dynamic control for wheeled mobile robots,” *Information Sciences*, vol. 179, no. 1–2, pp. 180–195, 2009.
- [44] G.-B. Huang and L. Chen, “Convex incremental extreme learning machine,” *Neurocomputing*, vol. 70, no. 16–18, pp. 3056–3062, 2007.
- [45] G.-B. Huang, L. Chen, and C.-K. Siew, “Universal approximation using incremental constructive feedforward networks with random hidden nodes,” *IEEE Transactions on Neural Networks*, vol. 17, no. 4, pp. 879–892, 2006.
- [46] D. T. Bui, P.-T. T. Ngo, T. D. Pham et al., “A novel hybrid approach based on a swarm intelligence optimized extreme learning machine for flash flood susceptibility mapping,” *Catena*, vol. 179, pp. 184–196, 2019.
- [47] Y. Yang, Y. Wang, and X. Yuan, “Parallel chaos search based incremental extreme learning machine,” *Neural Processing Letters*, vol. 37, no. 3, pp. 277–301, 2013.
- [48] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.
- [49] W. Huang, G. Song, H. Hong, and K. Xie, “Deep architecture for traffic flow prediction: deep belief networks with multitask learning,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.
- [50] D. Saif, S. M. El-Gokhy, and E. Sallam, “Deep belief networks-based framework for malware detection in Android systems,” *Alexandria Engineering Journal*, vol. 57, no. 4, pp. 4049–4057, 2018.
- [51] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A survey of deep neural network architectures and their applications,” *Neurocomputing*, vol. 234, pp. 11–26, 2017.
- [52] Y. W. Teh and G. E. Hinton, “Rate-coded restricted Boltzmann machines for face recognition,” in *Advances In Neural Information Processing Systems 13*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds., pp. 908–914, MIT Press, Cambridge, MA, USA, 2001.
- [53] G. E. Hinton, “A practical guide to training restricted Boltzmann machines,” in *Neural Networks: Tricks of the Trade*, G. Montavon, G. B. Orr, and K.-R. Müller, Eds., pp. 599–619, Springer, Berlin, Germany, Second edition, 2012.
- [54] M. M. Hassan, M. G. R. Alam, M. Z. Uddin, S. Huda, A. Almogren, and G. Fortino, “Human emotion recognition using deep belief network architecture,” *Information Fusion*, vol. 51, pp. 10–18, 2019.
- [55] G. E. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [56] S. P. Washington, M. G. Karlaftis, and F. L. Mannering, *Statistical and Econometric Methods for Transportation Data Analysis*, Chapman and Hall/CRC, Boca Raton, FL, USA, 2011.
- [57] F. J. Li, H. G. Han, and J. F. Qiao, “Pruning algorithm for extreme learning machine based on sensitivity analysis,” *Control and Decision*, vol. 29, pp. 1003–1008, 2014.
- [58] Z. Du, X. Li, L. Xi, and Q. Mao, “Improved sensitivity-analysis based pruning extreme learning machine,” *Control and Decision*, vol. 31, pp. 249–255, 2016.
- [59] L. Liu and R.-C. Chen, “A novel passenger flow prediction model using deep learning methods,” *Transportation Research Part C: Emerging Technologies*, vol. 84, pp. 74–91, 2017.