Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 5872225, 13 pages
https://doi.org/10.1155/2022/5872225

Research Article

Passenger Flow Prediction Using Smart Card Data from Connected Bus System Based on Interpretable XGBoost

Liang Zou,[1,2] Sisi Shu,[1] Xiang Lin,[1] Kaisheng Lin,[1] Jiasong Zhu,[1,2] and Linchao Li[1,2]

[1]*College of Civil and Transportation Engineering, Shenzhen University, Shenzhen 518060, China*
[2]*Institute of Urban Smart Transportation & Safety Maintenance, Shenzhen University, Shenzhen 518060, China*

Correspondence should be addressed to Linchao Li; lilinchao123@163.com

Received 6 September 2021; Accepted 4 December 2021; Published 7 January 2022

Academic Editor: Li Zhu

Bus passenger flow prediction is a critical component of advanced transportation information system for public traffic management, control, and dispatch. With the development of artificial intelligence, many previous studies attempted to apply machine learning models to extract comprehensive correlations from transit networks to improve passenger flow prediction accuracy, given that the variety and volume of traffic data have been easily obtained. The passenger flow on a station is highly affected by various factors such as the previous time step, peak hours or nonpeak hours, and extracting the key features from the data is essential for a passenger flow prediction model. Although the neural networks, *k*-nearest neighbor, and some deep learning models have been adopted to mine the temporal correlations of the passenger flow data, the lack of interpretability of the influenced variables is still a big problem. Classical tree-based models can mine the correlations between variables and rank the importance of each variable. In this study, we presented a method to extract passenger flow of different routes on the station and implemented a XGBoost model to find the contributions of variables to the prediction of passenger flow. Comparing to benchmark models, the proposed model can reach state-of-the-art prediction accuracy and computational efficiency on the real-world dataset. Moreover, the XGBoost model can interpret the predicted results. It can be seen that period is the most important variable for the passenger flow prediction, and so the management of buses during peak hours should be improved.

1. Introduction

Passenger flow prediction is important for advanced transportation information system (ATIS) and the planning for multimodal traffic management. A comprehensive classification of historical changing patterns of passenger flow for public bus stations is not only capable of finding the hot spots of public transportation of multimodal traffic network but also able to improve the accuracy of future passenger flow prediction. As a component of ATIS, accurate passenger flow prediction is the prerequisite for dynamic vehicle scheduling and public transport planning. The increasing need for short-term passenger flow prediction embedded in ATIS has led to a large amount of research on passenger flow prediction in the past ten decades. Before the breakthrough of artificial intelligence, passenger flow prediction models have been gradually changing from traditional statistical models to machine learning models. With the exponential growing of computational capability and the volume of traffic data, a large number of deep learning models, such as conventional neural network, recurrent neural network, and their extensions, have been adopted for short-term passenger flow prediction during recent years.

However, there are still some challenges in the prediction model. The first question is how to extract accurate passenger flow from the massive bus card data and vehicle operation data. The accurate data is the basic of the prediction model. Moreover, how much each variable contributes to prediction is also a question. There might be several routes at one station, and the buses of the same route could arrive many times during one predicted interval. The competition and complementation are critical factors contributing to the number of passenger flow. Considering the above two variables into current prediction models is necessary.

To address the above issues, we propose a new model for bus passenger flow prediction and to rank the influence factors. In this paper, we firstly presented the method to extract the passenger flow from the bus card data and the vehicle operation data. Then, XGBoost can be implemented to mine the temporal correlations in the time series data and find contributions of each variable. In order to learn the correlations between different routes of the same stations and fulfill passenger flow prediction, we take the number of buses arriving on the same routes and different routes into consideration in the prediction model. As tested on a real-world dataset collected from Guangzhou, the introduced preprocess method is effective and XGBoost achieves better predicted accuracy with strong interpretability comparing to existing benchmark models. The contributions of this paper are summarized as follows:

(1) An integral process to extract passenger flow from the raw data is presented. Moreover, a real-world dataset is used to test the proposed model

(2) The interaction between different routes could affect the passenger flow at a specific station. In this study, we consider the number of buses coming during the prediction interval to improve predicted accuracy and measure the influence

(3) Besides improving the accuracy, this study mines the factors from massive bus card data and vehicle operation data affecting passenger flow. The results can contribute to bus station and route planning

The rest of this paper is organized as follows. Section 2 discusses the existing literature. Section 3 defines XGBoost-related concepts and describes the methodology in detail. The data preprocessing procedures and evaluation criteria are presented in Section 4. The experimental results are shown in Section 5. Finally, we conclude the paper in Section 6.

2. Literature Review

During recent years, a large number of passenger flow prediction models for public transportation have been built. Generally, the prediction model can be divided into two categories: statistic-based models and machine learning-based models. The popular statistic-based models include autoregressive, moving average, and autoregressive moving average model. For example, it is found that Autoregressive Integrated Moving Average Model (ARIMA) has a high accuracy in predicting rail transit passenger flow [1]. Milenković et al. found a strong seasonal autocorrelation in time series and proposed a Seasonal Autoregressive Integrated Moving Average (SARIMA) method to predict railway passenger flow [2]. Tang et al. proposed a short-term passenger flow prediction framework and evaluated its performance with ARIMA, multiple linear regression, and support vector regression [3]. Zhang et al. built a two-step real-time prediction model, which first made rough prediction of bus passenger flow based on historical data, and then calibrated the rough prediction based on the extended Kalman filter (EKF) [4]. Gong et al. put forward a three-stage framework to predict the short-term passenger flow of bus stations. First, arrival passenger count (APC) is predicted based on SARIMA; then, departure passenger count (DPC) is predicted based on event algorithm, and finally, waiting passenger count (WPC) is predicted based on Kalman filter; the APC and DPC are used to update evolution functions in the third step [5]. To improve the reliability of subway passenger flow prediction, Li et al. proposed a hybrid model combining linear ARIMA model and nonlinear symbolic regression [6]; Ding et al. built their models based on joint ARIMA and generalized autoregressive conditional heteroscedasticity (GARCH) [7]. Wang et al. applied a hybrid model combining empirical pattern decomposition (EMD) and ARIMA to predict short-term traffic speeds on highways [8]. Some researchers combined ARIMA with other methods, such as bagging technique [9] and genetic programming [10], for improving prediction accuracy. However, with the increase of the data, the statistic-based models have some limitations. Firstly, the statistic-based models have some strong assumptions which might not be relevant to traffic data, so it is difficult to mine the useful features. Moreover, statistic-based models are not good at handling categorical variables in the massive data.

The development of machine learning models gives us some new opportunities to improve passenger flow prediction. For example, Liu et al. designed a new deep learning architecture called modular convolutional neural network based on the experiment of decision tree-based models, to solve the large-scale bus passenger flow prediction problem [11]. Liu and Chen proposed an unsupervised training model based on the combination of stacked autoencoder (SAE) and deep neural network (DNN) to forecast hourly bus passenger flow [12]. Many researchers have studied the subway passenger flow forecast. Li et al. introduced a new dynamic radial basis function neural (RBF) network with dynamic input to predict the outbound passenger flow in subway stations [13]. Zhang et al. combined with Residual Network (ResNet), Graphic Convolutional Network (GCN), and Long and Short-Term Memory (LSTM) put forward a deep learning architecture [14]. Zhao et al. proposed a new three-stage framework based on a hierarchical clustering algorithm (AHC) and tree-based models to select the appropriate feature variables [15]. They proposed a hybrid spatial and temporal deep learning neural network (HSTDL-NET) [16]. Liu et al. developed an end-to-end deep learning architecture based on recurrent neural network (RNN) and LSTM [17]. Hao *et al.* proposed an end-to-end deep learning framework based on LSTM network to predict the number of passengers getting off at each station in the last few short-term periods [18]. Li et al. proposed a multi-station passenger flow prediction method for subway stations, using a dynamic weight combination of Support Vector Machines (SVM) and RBF to improve the stability of the prediction model [19]. Zhang et al. proposed a channel-wise attentive split–convolutional neural network (CAS-CNN) to predict short-term OD flow. This is the first time that the split convolutional neural network is applied to short-term OD prediction in urban rail transit [20]. However,

these models have been criticized for their poor interpretability and the need for a great deal of computing resource.

The ensemble tree method was developed to solve the above deficiencies and has been widely applied during recent years. For example, Xu et al. [21] and Liu et al. [22] built a Gradient Boosting Decision Tree (GBDT) model to predict the bus passenger flow, so as to make the prediction more accurate. Zhang *et al.* applied the LightGBM model to predict the subway passenger flow, taking into account the influence of the transfer passenger flow on the prediction [23]. Liu et al. established a passenger flow forecasting model by using Random Forest [24]. In order to further improve the accuracy and efficiency of passenger flow forecasting, researchers combine other methods with ensemble tree model. A new model combining singular spectrum analysis with AdaBoost-weighted extreme learning machine is proposed [25]. Lin and Tian introduced a hybrid model combining Random Forest and LSTM to predict subway passenger flow [26]. Dong *et al.* proposed a traffic flow prediction model combining wavelet decomposition and reconstruction with the Extreme Gradient Boosting (XGBoost) algorithm [27]. Du et al. used the combined model of XGBoost and LSTM in the short-term traffic prediction of the base station [28]. Yun et al. built a local optimal fusion model based on LSTM, LightGBM, and dynamic regression device [29]. Wang *et al.* took Multivariable Linear Regression (MLR), *k*-nearest neighbor (KNN), XGBoost, and Gated Recurrent Unit (GRU) as four seed models to establish a regression integration model to accurately predict short-term passenger flows of urban public transport [30]. The aim of this paper is to develop a prediction using the massive bus card and bus operation data and find the importance of variables for the prediction accuracy, and therefore, XGBoost is implemented.

3. Methodology

In 2015, Chen et. al proposed an improved integrated learning algorithm called Extreme Gradient Boosting tree (XGBoost) [31]. Compared with other gradient boosting algorithms, XGBoost has a significant improvement in accuracy and speed for regression and classification problems, because the model can integrate multiple regression trees to make decisions under the framework of boosting. In this section, the theory of XGBoost will be introduced.

3.1. Regression Tree. Regression tree is a binary tree to solve regression problems. Assuming the built regression tree has T leaf nodes, so the regression tree divides the input space into T units as $R_1, R_2, \cdots, R_T$. The formulation of regression tree model can be expressed as

$$f(x) = \sum_{j=1}^{T} w_j I\left(x \in R_j\right), \tag{1}$$

where x is the sample; $f(x)$ is the prediction value; $I(x)$ is the identify function which will return 1 if x is in the subset R_j; and w is the output value on the divided unit. To divide the input space best, the square error can be used to represent the prediction error of the training data which can be expressed as

$$\text{Square error} = \sum_{x_i \in R_j} \left(y_i - f(x_i)\right)^2, \tag{2}$$

where y_i is the ith predicted value; to minimize the overall error of regression tree, it is only necessary to set the predicted value in each unit area to the mean value of the output of the sample set contained in the area:

$$\widehat{w}_j = \text{ave}\left(y_i \mid x_i \in R_j\right), \tag{3}$$

where $\text{ave}(x)$ is the function to obtain the average value of x. The details of the optimal procedure are described as below. Firstly, in the input space of the training data set, the dth feature $x^{(d)}$ and the corresponding value s is selected for each division. After dividing into two subregions, the samples with feature values less than s are divided into the left subtree, and the samples with feature values greater than s are divided into the right subtree:

$$\begin{aligned} R_1(d, s) &= \left\{x \mid x^{(d)} \le s\right\}, \\ R_2(d, s) &= \left\{x \mid x^{(d)} > s\right\}. \end{aligned} \tag{4}$$

Then, all input variables are traversed to find the optimal segmentation feature j and the optimal segmentation point s, constituting one pair (d, s), by minimizing the loss function of the two subregions:

$$\min_{d,s}\left[\min_{c_1} \sum_{x_i \in R_1(d,s)} \left(y_i - w_1\right)^2 + \min_{c_2} \sum_{x_i \in R_2(d,s)} \left(y_i - w_2\right)^2\right]. \tag{5}$$

Finally, the output value of each subregion can be determined by

$$\begin{aligned} \widehat{w}_1 &= \text{ave}(y_i \mid x_i \in R_1(d, s)), \\ \widehat{w}_2 &= \text{ave}(y_i \mid x_i \in R_2(d, s)). \end{aligned} \tag{6}$$

The above process for the division of the input space is repeated until the division cannot be continued.

3.2. The Integration Process of the XGBoost Model. Boosting is an additive model and one of the methods for ensemble learning which complete learning tasks by constructing and combining multiple learners. There is a strong dependency between these individual learners, which must be generated serially, and each weak learner must be upgraded to a strong learner to reduce the bias of the model. XGBoost is one of the tree-based ensemble learning which grows a tree by continuously adding regression tree and splitting features. Each time, a tree is added, and the model learns a new function to fit the residual of the previous tree

prediction; that is, the tth regression tree is obtained by training based on the model of the previous $t-1$ iterations. After the training is completed and t trees are obtained, the predicted value of each sample is obtained by falling into the corresponding leaf node of each tree according to the characteristics of the sample. The model obtained after the tth iteration is

$$\begin{cases} \widehat{y}_i^{(0)} = 0, \\ \widehat{y}_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = \widehat{y}_i^{(t-1)} + f_t(x_i), \end{cases} \tag{7}$$

where $\widehat{y}_i^{(t)}$ is the predicted value of the ith sample in the tth round of the model, $\widehat{y}_i^{(t-1)}$ is the score of the ith sample in the model retained for the previous $t-1$ round, and $f_t(x_i)$ is the score of the ith sample newly added to the regression tree. The detail of the model for passenger prediction is introduced in the following part.

Assuming that the XGBoost model integrates K trees, the predicted value of each sample is

$$y_i = \sum_{k=1}^{K} f_k(x_i), \quad f_k \in \mathcal{F}, \tag{8}$$

where $\mathcal{F}$ is the set of all regression trees and f is one of the regression trees in $\mathcal{F}$. The integration process of the XGBoost model is illustrated in Figure 1.

In the passenger flow prediction problem, the goal is to find a suitable model and continuously optimize the parameters to minimize the difference between prediction and observation. Therefore, this paper defines the objective function of the XGBoost model and minimizes the objective function to find the best parameters.

The objective function of the XGBoost model in this paper is expressed as

$$\text{obj} = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t)}\right) + \sum_{k=1}^{K} \Omega(f_k). \tag{9}$$

The objective function has two parts. The first part $\sum_{i=1}^{n} l(y_i, \widehat{y}_i^{(t)})$ is the loss function, which measures the fitting effect of the model on the training set. The smaller the value of the loss function, the better the fitting effect. The latter part $\sum_{k=1}^{K} \Omega(f_k)$ is the regular term, which measures the complexity of the model. Optimizing the regular term can avoid the weak generalization ability. The regular term $\Omega(f_k)$ is expressed as

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2, \tag{10}$$

where T is the number of leaf nodes of the tree, w_j is the score of the jth leaf node, γT represents the complexity of the number of leaf nodes of the tree, $\lambda \sum_{j=1}^{T} w_j^2$ represents the regular term of $L2$, and γ is the coefficient that controls the number of leaf nodes.

The objective function of the XGBoost model obtained after the tth iteration is expressed as

$$\begin{aligned} \text{obj}^{(t)} &= \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t)}\right) + \sum_{k=1}^{t} \Omega(f_k) = \sum_{i=1}^{n} l\left(y_i, \widehat{y}_i^{(t-1)} + f_t(x_i)\right) \\ &\quad + \Omega(f_t) + \text{constant}, \end{aligned} \tag{11}$$

where constant is the regularization penalty term of the previous $t-1$ iterations.

After performing the second-order Taylor expansion on the error term of the objective function, the algorithm can be updated as

$$\begin{aligned} \text{obj}^{(t)} &\approx \sum_{i=1}^{n} \left[l\left(y_i, \widehat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] \\ &\quad + \Omega(f_t) + \text{constant}, \end{aligned} \tag{12}$$

where g_i is the first derivative of the error function and h_i is the second partial derivative of the error function. The expressions are shown as follows:

$$\begin{aligned} g_i &= \partial_{\widehat{y}_i^{(t-1)}} l\left(y_i, \widehat{y}_i^{(t-1)}\right), \\ h_i &= \partial^2_{\widehat{y}_i^{(t-1)}} \left(y_i, \widehat{y}_i^{(t-1)}\right). \end{aligned} \tag{13}$$

Since the error term and regular term of the previous $t-1$ iterations are constant terms, they have no effect on the optimization of the objective function of the tth iteration, so they can be omitted. The simplified objective function of the tth iteration is

$$\text{obj}^{(t)} \approx \sum_{i=1}^{n} \left[g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i) \right] + \Omega(f_t). \tag{14}$$

In regression trees, the score of the sample will eventually fall into one leaf node, so it can be a collection of all samples of the same node. The mapping function of regression tree can be transformed into

$$f_t(x_i) = w_q(x_i), \quad w \in R^T, q : R^d \longrightarrow \{1, 2, \cdots, T\}, \tag{15}$$

where w_q represents the sum of all sample scores of the qth leaf node.

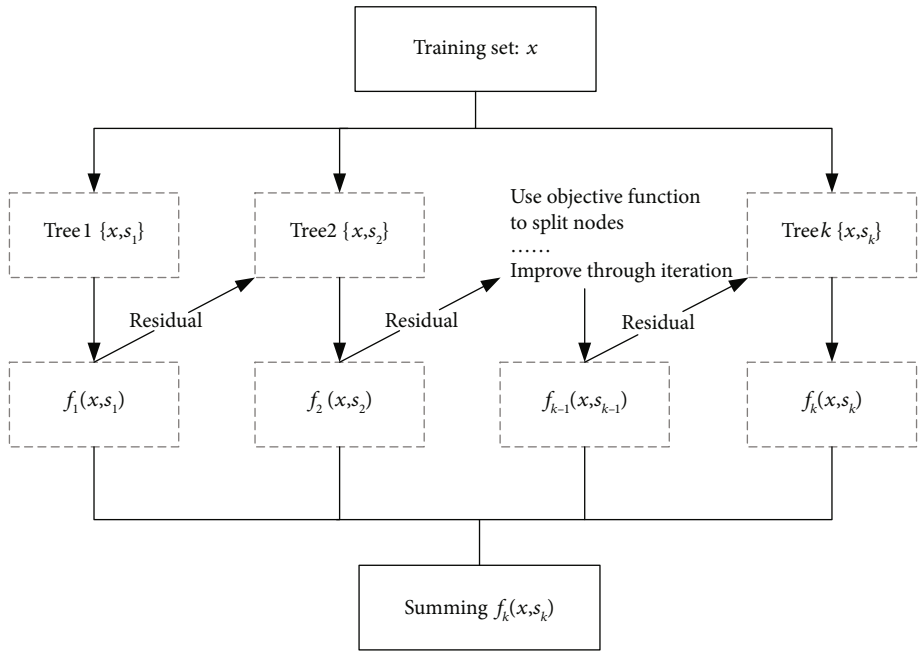


Figure 1: The integration process of the XGBoost model.

So far, the objective function can be expressed as follows:

$$\begin{aligned}\text{obj}^{(t)} &\approx \sum_{i=1}^{n}\left[g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t)\\ &= \sum_{i=1}^{n}\left[g_i w_q(x_i) + \frac{1}{2}h_i w_q^2(x_i)\right] + \gamma T + \frac{1}{2}\lambda\sum_{j=1}^{T} w_j^2\\ &= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma T\\ &= \sum_{j=1}^{T}\left[G_j w_j + \frac{1}{2}\left(H_j + \lambda\right)w_j^2\right].\end{aligned} \tag{16}$$

Some definitions of the above formula are

$$\begin{aligned}G_j &= \sum_{i\in I_j} g_i,\\ H_j &= \sum_{i\in I_j} h_i.\end{aligned} \tag{17}$$

Through the above formula, the problem of minimizing the objective function is transformed into a problem of finding the extreme value of a quadratic equation of one variable about the fraction w of leaf nodes. Therefore, the best score of the leaf nodes is

$$w_j^* = -\frac{G_j}{H_j + \lambda}. \tag{18}$$

Then, the objective function corresponding to this best score is

$$\text{obj}^{(t)} = -\frac{1}{2}\sum_{j=1}^{T}\frac{G_j^2}{H_j + \lambda} + \gamma T. \tag{19}$$

Equation (19) is the score of the tree structure in which $\text{obj}^{(t)}$ is a function for scoring the tree structure. The lower the score, the better the tree structure.

When the XGBoost model traverses all the feature points and splits the tree nodes, it uses the above objective function as the evaluation function. If the total value of the objective function of the left and right subtrees after the split is increased compared with the original and the increase value is greater than a certain threshold, the split point that can obtain the maximum value of the objective function would be found continuously. If the point does not exist, no split is performed. This maximum value of the objective function is the gain. In t iterations, when splitting a leaf node, the gain before and after splitting is defined as

$$\text{Gain} = \frac{1}{2}\left(\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right) - \gamma, \tag{20}$$

where the first part in brackets is the score of the left subtree after the split, the second part in brackets is the score of the right subtree after the split, and the third part in brackets is the score before the split. γ represents the cost value of the complexity by adding a leaf node. The larger the value of Gain, the lower the value of the objective function after splitting, and the better the tree structure.

4. Data and Evaluation Criteria

4.1. The Format of Raw Data. In this study, the bus card data and vehicle operation data of 30 bus stations in Guangzhou

from April 24 to May 20, 2018, were selected to evaluate the performance of the model. The formats of bus card data and vehicle operation data are listed in Tables 1 and 2, respectively. From the raw data, the passenger flow cannot be obtained directly. Therefore, we need to match the bus card data with the vehicle operation data. In this paper, bus card data and vehicle operation data are matched using vehicle ID number. The format of the matched data is shown in Table 3.

4.2. The Preprocess of Data. From the raw dataset, we cannot get the number of passengers at a specific station during a prediction interval (30 min in this study), because the bus card data cannot reflect the actual arrival time of passengers. To obtain the real passenger flow data as much as possible, the arrival of passengers was assumed to follow a uniform distribution pattern according to previous studies. On the basis of this assumption, the number of passengers boarding the vehicle at a station is evenly distributed from the departure of the previous bus to the arrival of the current bus. The specific process of passenger flow calculation is as follows:

Input: the specific route ID, the site ID, and the matched data in Table 3

Output: the short-term passenger flow of the site

Process: according to the data of the selected station in the selected bus route, find out the data corresponding to different bus work shifts, get the interval time of different bus work shifts and the number of passengers, distribute the number of passengers evenly to the interval time, and then, get short-term passenger flow data after accumulation. For the convenience of analysis, the prediction interval T of short-term passenger flow is set to 30 minutes. The process of passenger flow calculation is shown in Figure 2. According to the above calculation process, a total of 756 short-term passenger flow data for each route at a single station from 7 am to 9 pm every 30 minutes are calculated.

4.3. Evaluation Criteria. Mean absolute percentage error (MAPE), root mean square error (RMSE), and mean absolute error (MAE) have been commonly used as evaluation criteria in previous studies that can measure the quality of prediction models. The calculated time (TIME) is an evaluation index to measure the computational efficiency of prediction models.

The formulas of MAPE, RMSE, MAE, and TIME are as follows:

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\widehat{y}_i - y_i}{\widehat{y}_i}\right|, \tag{21}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y\wedge_i - y_i)^2}, \tag{22}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\widehat{y}_i - y_i|, \tag{23}$$

$$\mathrm{TIME} = T_{\mathrm{end}} - T_{\mathrm{start}}, \tag{24}$$

where n is the number of samples, $\widehat{y}_i$ is the true value, y_i is the predicted value, and T_{start} and T_{end} represent the time when the model calculation starts and ends, respectively. The range of MAPE is [0, +∞). When the value of MAPE is 0%, it means there is no error. Root mean square error (RMSE) reflects the magnitude of the deviation between the predicted value and the true value. The unit of TIME is second.

Table 1: The format of the bus card data.

Name	Description
Bus_no	Vehicle ID number
Cardtype	The type of card
Lineno	Bus line number
Logiccardno	Card ID
Tim	Bus card trading hours

Table 2: The format of the bus operation data.

Name	Description
ID	Vehicle itinerary ID
OBUID	Vehicle ID number
ROUTE_ID	Bus route ID
ROUTE_CODE	Bus route code
ROUTE_NAME	Bus route name
SERVICE_NUMBER	Up or down
TRIP_ID	Bus work shifts ID
BUS_STOP_CODE	Bus stop code
ROUTE_STA_ID	The ID of the bus stop
STATION_ID	The ID of the route station
STATION_NAME	The name of the bus stop
AD_FLAG	Inbound or outbound
AD_TIME	Inbound or outbound time

Table 3: The format of the matched data.

Name	Examples
Logiccardno	5100000326361642
Tim	20180731000000
BUSID	1805210000069477
OBUID	989416
ROUTE_ID	765
SERVICE_NUMBER	0
TRIP_ID	001
BUS_STOP_CODE	211203
ROUTE_STA_ID	89500
STATION_ID	100004932
STATION_NAME	Hengjiao Station
AD_FLAG	0
AD_TIME	2018-05-21 00:00:06

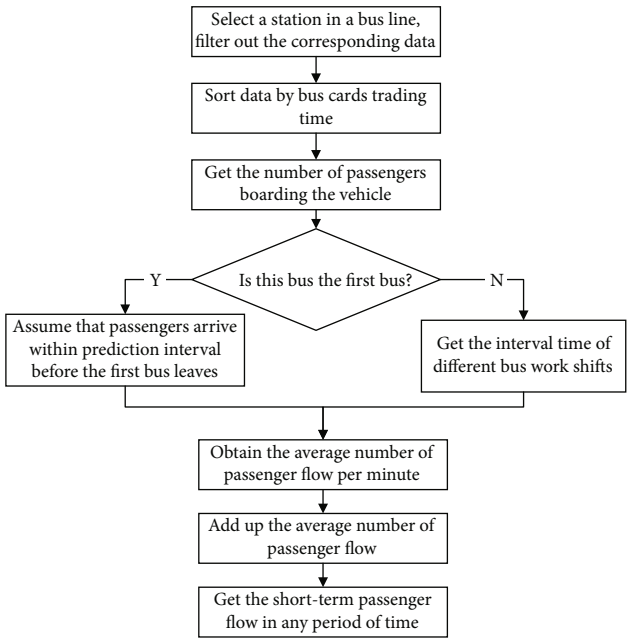


Figure 2: The process of passenger flow calculation.

5. Results

5.1. Optimization of the Prediction Model

5.1.1. Input Variable Selection. Period. The distributions of the short-term passenger flow of buses are shown in Figure 3. It can be seen that the passenger flow of most routes is higher in the peak hours than nonpeak hours. The passenger flow of bus changes dynamically for one day; however, it shows a similar change between workday and weekend. The change of passenger flow in the weekend is different from that of workdays. It has also been proved by previous studies. Therefore, weekend or workday is also taken into consideration in the prediction model.

Passenger flow of previous period. The closest thing to the current state of passenger flow is the passenger flow of the adjacent previous period, and the short-term passenger flow of bus is related to the number of passenger flow of the adjacent previous period.

The number of buses arriving on the predicted routes during the prediction interval. According to the calculation of passenger flow, it can be seen that there are fluctuations in the short-term passenger flow of conventional buses. In a prediction interval, the number of buses arriving of the same routes is higher than in the past and also is more than different considered routes, so the probability of passengers taking this route is higher than in the past; that is, the passenger flow on this route increases; otherwise, it decreases.

The number of buses arriving on the other routes during the prediction interval. In this article, some routes that have successively repeated sites on a route are called different routes into consideration. In real life, passengers usually consider the different routes that arrive at the station first when taking a bus. The more buses on the different routes into consideration arrive, the more passenger flow will be divided, and the smaller the passenger flow of the select route.

In the end, period, workday or weekend, passenger flow of previous period, the number of buses arriving on the predicted routes during the prediction interval, and the number of buses arriving on the other routes during the prediction interval are selected as the feature input of the conventional short-term passenger flow prediction model based on the XGBoost model.

5.1.2. Model Parameter Settings. The parameters of the XGBoost model include general parameters, lifting parameters, and learning task parameters. The general parameters control the overall function. The learning task parameters guide the model to perform optimization. The promotion parameters control the regression tree at each step. The general parameters and learning task parameters of the XGBoost model include *booster*, *objective*, and *eval_metric* as shown in Table 4. The goal of predicting passenger flow is a regression problem, and so the base learner of the XGBoost model used in this article is a regression tree. RMSE is used to evaluate the accuracy of the model.

The lifting parameters are selected using a grid search method. Take the process of adjusting the parameter *n_estimators* as an example. The candidate values of *n_estimators* are 200, 300, 400, 500, and 600. Firstly, we change the value of *n_estimators*, while the other parameters remain unchanged, and we use cross-validation and RMSE to measure the performance of models. Then, the best parameter value is selected. According to RMSE, the optimal *n_estimators* is 300. According to the above steps, the following parameters of a forecast model of this route are selected in turn: *min_child_weight*, *max_depth*, *gamma*, *subsample*, *subsample_bytree*, *reg_alpha*, *reg_lambda*, *learning_rate*. The optimal values are shown in Table 5.

After adjusting all the parameters, the construction of the short-term passenger flow prediction model for conventional public transportation based on the XGBoost model has been completed.

5.2. Comparison with Traditional Model. Currently, there are many prediction models for short-term passenger flow of conventional public transportation. Among them, the models that can perform multivariable short-term passenger flow prediction mainly include KNN, BP neural network, and LSTM. To show the better prediction effect of the proposed model, KNN, the BP neural network, and LSTM are established as benchmark models. Meanwhile, an XGBoost predicted model without the number of buses arriving is also implemented.

5.2.1. KNN Regression Model. Use *sklearn.neighbors.KNeighborsRegressor* in Python to build a short-term passenger flow prediction model for conventional public transportation based on KNN regression. In the KNN regression model, the key parameters are *n_neighbor* and *weight*. Take No. 125 bus at the station of Zhongshan 8th Road Station Substation 1 as an example, and finally, get the value of *n_neighbor* as 5 and *weight* as *uniform* through training.

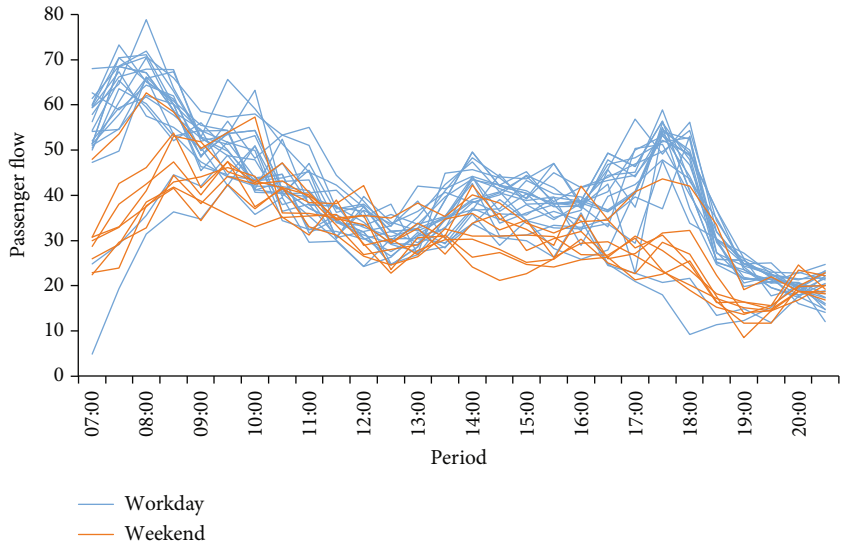


Figure 3: Time distribution of short-term passenger flow.

Table 4: Selection of general parameters and learning task parameters.

Parameters	Value
Booster	gbtree
Objective	reg: linear
eval_metric	RMSE

Table 5: The best value of each parameter.

Parameters	Value
Learning_rate	0.01
n_estimators	300
max_depth	8
min_child_weight	4
Subsample	0.8
colsample_bytree	0.8
Gamma	0.1
reg_alpha	1
reg_lambda	1

5.2.2. BP Neural Network Model. The *keras* deep learning framework is used to build a short-term passenger flow prediction model for conventional public transportation based on BP neural network. Take the 125 bus at the substation of Zhongshan 8th Road Station as an example. After repeated experiments, the number of input layer nodes of the BP neural network model is selected as 70, the number of hidden layer nodes is 30, and the number of output layer nodes is 1; that is, the model structure is 70-30-1. Among them, the activation function of the hidden layer and the output layer is selected as the *relu* function. In addition, the number of iterations of the BP neural network is selected as 200, and the learning rate is 0.01.

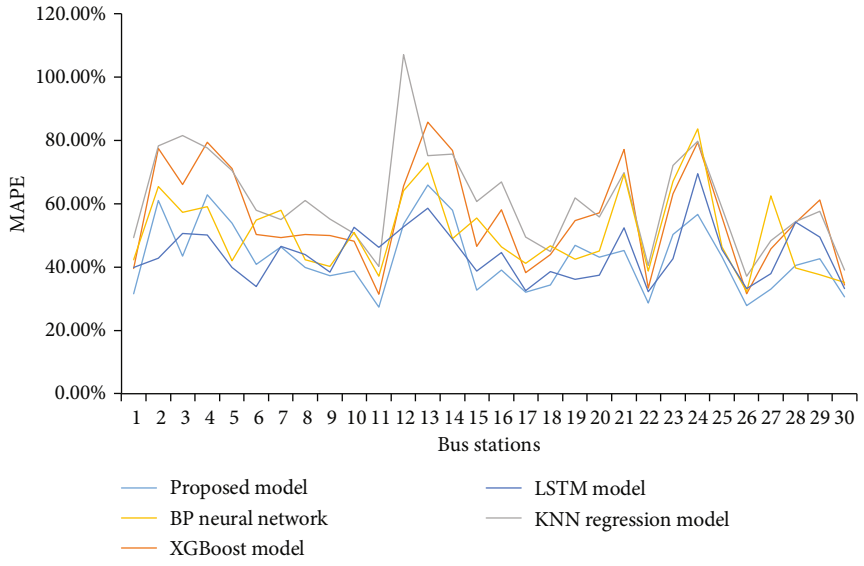
Table 6: The best value of each parameter of XGBoost without the number of buses arriving.

Parameters	Value
Learning_rate	0.01
n_estimators	350
max_depth	8
min_child_weight	4
Subsample	0.8
colsample_bytree	0.8
Gamma	0.1
reg_alpha	1
reg_lambda	1

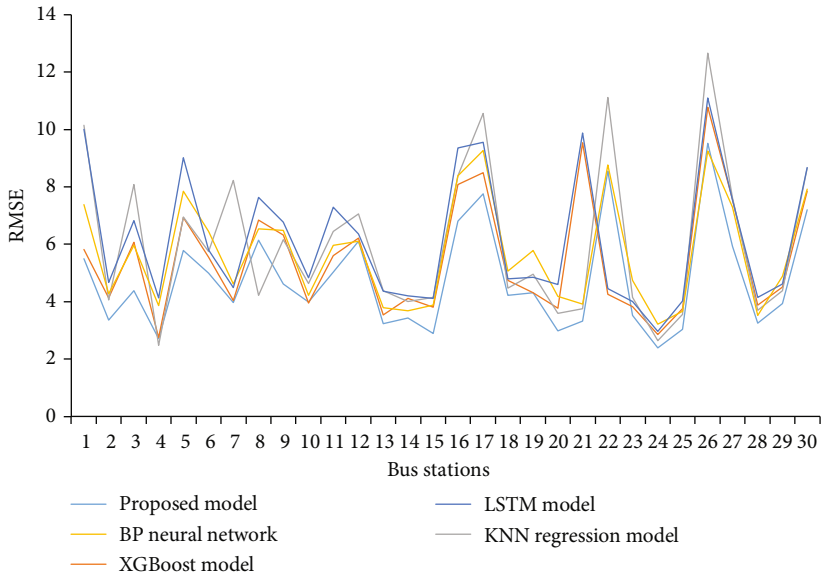
5.2.3. LSTM Model. The *keras* deep learning framework is used to build a short-term passenger flow prediction model for conventional public transportation based on LSTM. Take the 125 bus at the substation of Zhongshan 8th Road Station as an example. After repeated trials, the number of input layer nodes of the selected LSTM model is 20, the number of hidden layer nodes is 15, the number of output layer nodes is 1, the step size is 4, the activation function of the hidden layer and the output layer is the *relu* function, and the optimizer is *adam*. In addition, the number of iterations of LSTM is 50 times.

5.2.4. XGBoost Model without the Number of Buses Arriving. Taking the 125 bus at the substation of Zhongshan 8th Road Station as an example, the best values of its parameters are shown in Table 6.

The MAPE, RMSE, and MAE of the above prediction models for each bus route are shown in Figure 4. It can be clearly seen that, regardless of indicators MAE, MAPE, or RMSE, the proposed model has the lowest value among most stations. For a quantitative analysis of the prediction

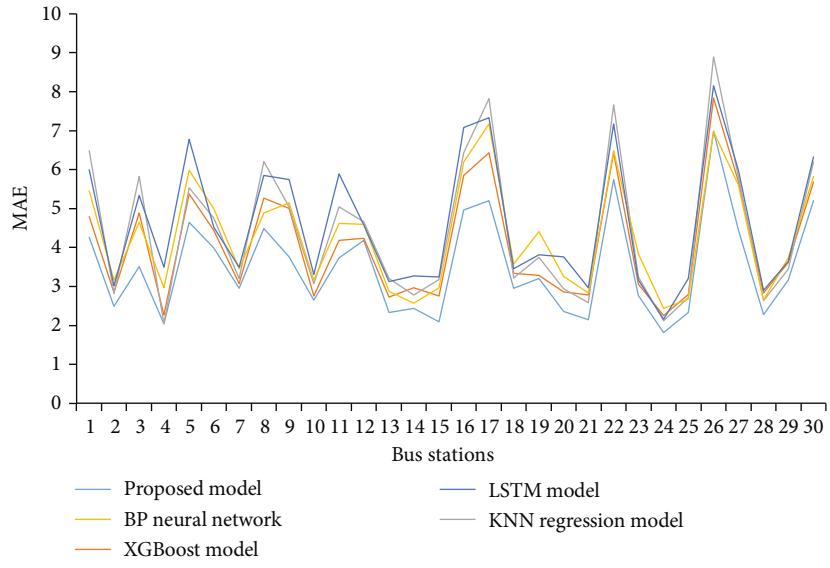


(a)



(b)

Figure 4: Continued.



(c)

Figure 4: (a) MAPE results of 30 bus stations in five models. (b) RMSE results of 30 bus stations in five models. (c) MAE results of 30 bus stations in five models.

Table 7: Comparison of the average value of the evaluation indicators of 30 bus stations.

Model	MAPE (%)	RMSE	MAE	Time (s)
The proposed model	42.90	4.76	3.53	10.63
XGBoost model without the number of buses arriving	55.86	5.46	4.08	—
KNN regression model	61.08	6.03	4.44	11.78
BP neural network model	50.85	5.70	4.31	40.21
LSTM model	44.11	6.17	4.63	28.15

results, Table 6 shows the comparison of the average values of the evaluation indicators of five prediction models.

From Table 7, the proposed model has higher accuracy than other models, and the number of buses arriving as a feature input can improve the prediction effect of the XGBoost model. Among them, the accuracy of the proposed model is better than that of the KNN regression model and the BP neural network model, and it is significantly better than the XGBoost model without the number of buses arriving.

The calculation time of the XGBoost model is slightly shorter than that of the KNN regression model, and it is much lower than the calculation time of the BP neural network model and the LSTM model. It is concluded that the XGBoost model is faster than the KNN regression model, BP neural network model, and LSTM model.

5.3. Robustness of the Model. To evaluate the robustness, we compare the performance of the proposed model.

5.3.1. Analysis by Time Period. Firstly, the accuracy of the proposed model is compared under different time periods: peak hours and nonpeak hours. The results are shown in Table 8. It can be seen that the MAPE of the proposed model during peak hours is significantly better than that during low peak periods. This may be caused by its relatively stable passenger flow distribution during peak hours which has less volatility and less influence by various factors. But at the same time, the absolute error (RMSE and MAE) in the low-peak period is relatively low. This is because the passenger flow in the low-peak period is much smaller than that in the peak period. It can be seen that the proposed model has a good prediction effect during peak hours.

Table 8: Comparison of the average value of the evaluation index under the peak hours and the nonpeak hours.

Type of time period	MAPE (%)	RMSE	MAE
Peak hours	30.26	9.98	8.9
Nonpeak hours	43.9	3.76	2.95

Table 9: Comparison of the average value of evaluation indicators under different passenger flow types.

Passenger flow type	MAPE (%)	RMSE	MAE
Unimodal type	45.59	4.55	3.48
Bimodal type	43.26	4.68	3.42
Other types	39.85	5.07	3.69

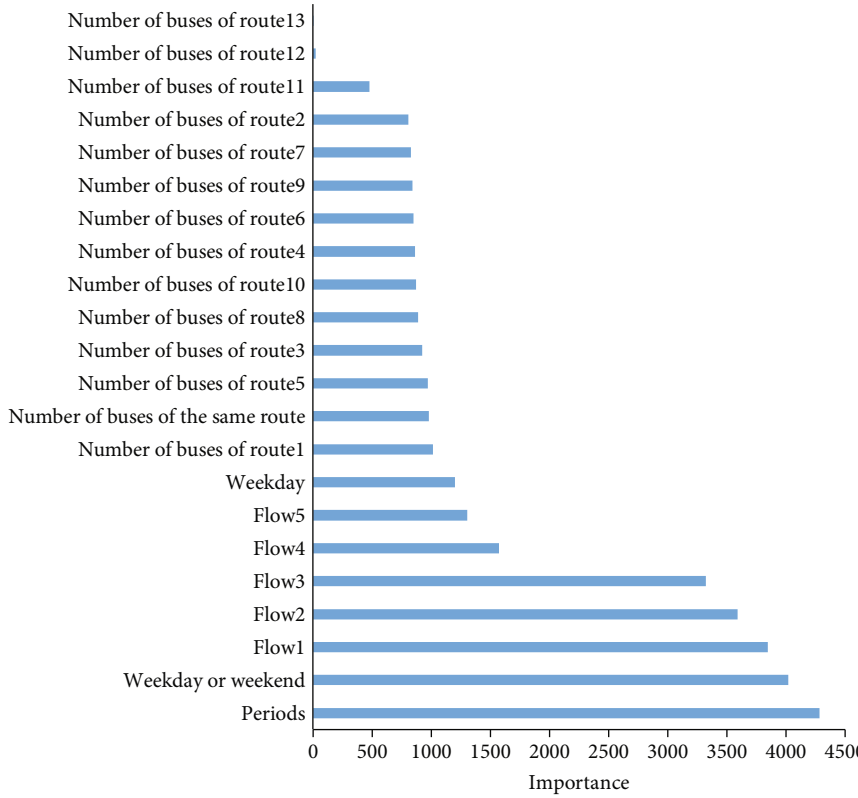


Figure 5: Feature importance score of the prediction model of 30 routes.

5.3.2. Analysis by the Type of Passenger Flow Distribution. Secondly, the accuracy of the proposed model is compared under different distributions. The distributions of passenger flow are divided as unimodal type, bimodal type, and other type. The comparison results are shown in Table 9.

It can be seen that in the proposed model for the three types of passenger flow distribution, the prediction accuracy (MAPE) of the other types is better than that of the single-peak and double-peak types. This may be due to the fact that other types include multipeak passenger flow distribution types, which have more peak hours, and the passenger flow distribution during peak hours with less volatility is relatively stable and has been less affected by various factors. In general, other types of prediction accuracy are higher. The absolute error (RMSE and MAE) of other types is lower, which may be due to the fact that the average short-term passenger flow of other types is higher than that of the single-peak type and the double-peak type.

5.4. Influence Degree of Variables. For different stations, different XGBoost prediction models are built. We have analyzed the weight of each variable contributing to the passenger flow prediction. It can be found that the contribution of variables of different routes is similar, and so we use the average of the routes to take further analysis. The result is shown in Figure 5. It can be seen that period contributes the most among all variables and following is the date type indicating the temporal correlation is strong in passenger flow data. Interestingly, the contribution of buses from other routes is greater than that of buses from the same route, indicating the competition between different routes is fierce. Also, the importance of other routes for the prediction is different. To mine the relationship between different routes should be worthy of study and concerns in the future work.

6. Conclusion

In this paper, we propose an ensemble tree method XGBoost to predict passenger flow of bus routes. Since the passenger flow of a station is highly influenced by the competition and complementation of other routes and buses of the same route, we take the number of routes and the number of buses during the predicted interval into the model to improve the accuracy. Comparing to the model, which does not consider the above two factors, the MAPE, RMSE, and MAE can be improved by 30.21%, 14.71%, and 15.58%, respectively. Moreover, the proposed model was compared to some benchmark models using the same data from Guangzhou; the results show it can achieve superior prediction performance. Surprisingly, XGBoost, consuming less computing resource than deep learning model LSTM, can achieve higher accuracy.

Also, the proposed model has stronger reliability and interpretability than other benchmark models. We have evaluated the model under different passenger flow types and periods, and the proposed model can yield stable results. Moreover, the weights of variables contributing to passenger flow prediction are analyzed. It can be concluded that temporal information is critical to the prediction model. The

competition of the different routes considered in the prediction model could improve the accuracy.

In the future, we will take more variables, such as spatial variables and daily variables, into consideration in the model to further improve the accuracy. Also, we will work on extracting the passenger flow during shorter interval to build short-term prediction model.

Data Availability

The data used to support the findings of this study have not been made available because of privacy.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This paper is supported by Shenzhen Science and Technology Innovation Committee (Grant No. JCYJ2017 0818142947240), Foundation for Distinguished Young Talents in Higher Education of Guangdong, China (Grant No. 2019KQNCX126), and Science and Technology Planning Project of Guangdong Province under (Grant No. 2018B020207005).

References

[1] S. Liu, S. Liu, Y. Tian, Q. Sun, and Y. Tang, "Research on forecast of rail traffic flow based on ARIMA model," *Journal of Physics: Conference Series*, vol. 1792, no. 1, p. 012065, 2021.

[2] M. Milenković, L. Švadlenka, V. Melichar, N. Bojović, and Z. Avramović, "SARIMA modelling approach for railway passenger flow forecasting," *Transport*, vol. 33, no. 5, pp. 1–8, 2016.

[3] L. Tang, Y. Zhao, J. Cabrera, J. Ma, and K. L. Tsui, "Forecasting short-term passenger flow: an empirical study on Shenzhen Metro," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, pp. 3613–3622, 2018.

[4] J. Zhang, D. Shen, L. Tu et al., "A real-time passenger flow estimation and prediction method for urban bus transit systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3168–3178, 2017.

[5] M. Gong, X. Fei, Z. H. Wang, and Y. J. Qiu, "Sequential framework for short-term passenger flow prediction at bus stop," *Transportation Research Record*, vol. 2417, no. 1, pp. 58–66, 2014.

[6] L. Li, Y. Wang, G. Zhong, J. Zhang, and B. Ran, "Short-to-medium term passenger flow forecasting for metro stations using a hybrid model," *KSCE Journal of Civil Engineering*, vol. 22, no. 5, pp. 1937–1945, 2018.

[7] C. Ding, J. Duan, Y. Zhang, X. Wu, and G. Yu, "Using an ARIMA-GARCH modeling approach to improve subway short-term ridership forecasting accounting for dynamic volatility," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 4, pp. 1054–1064, 2017.

[8] H. Wang, L. Liu, Z. S. Qian, H. Wei, and S. Dong, "Empirical mode decomposition-autoregressive integrated moving average," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2460, no. 1, pp. 66–76, 2014.

[9] S. Shahriari, S. A. Milad Ghasri, and T. R. Sisson, "Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction," *Transportmetrica A: Transport Science*, vol. 16, no. 3, 2020.

[10] C. Xu, Z. Li, and W. Wang, "Short-term traffic flow prediction using a methodology based on autoregressive integrated moving average and genetic programming," *Transport*, vol. 31, no. 3, pp. 343–358, 2016.

[11] Y. Liu, C. Lyu, X. Liu, and Z. Liu, "Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 4, pp. 1–10, 2020.

[12] L. Liu and R.-C. Chen, "A novel passenger flow prediction model using deep learning methods," *Transportation Research Part C*, vol. 84, pp. 74–91, 2017.

[13] H. Li, Y. Wang, X. Xinyue, L. Qin, and H. Zhang, "Short-term passenger flow prediction under passenger flow control using a dynamic radial basis function network," *Applied Soft Computing Journal*, vol. 83, p. 105620, 2019.

[14] J. Zhang, F. Chen, and Q. Shen, "Cluster-based LSTM network for short-term passenger flow forecasting in urban rail transit," *Transit*, vol. 7, pp. 147653–147671, 2019.

[15] L. Yangyang Zhao, Z. M. Ren, and X. Jiang, "Novel three-stage framework for prioritizing and selecting feature variables for short-term metro passenger flow prediction," *Transportation Research Record*, vol. 2674, no. 8, pp. 192–205, 2020.

[16] H. Zhang, J. He, J. Bao, Q. Hong, X. Shi, and G. E. Cantarella, "A hybrid spatiotemporal deep learning model for short-term metro passenger flow prediction," *Journal of Advanced Transportation*, vol. 2020, 12 pages, 2020.

[17] Y. Liu, Z. Liu, and R. Jia, "DeepPF: a deep learning based architecture for metro passenger flow prediction," *Transportation Research Part C*, vol. 101, pp. 18–34, 2019.

[18] S. Hao, D.-H. Lee, and D. Zhao, "Sequence to sequence learning with attention mechanism for short-term passenger flow prediction in large-scale metro system," *Transportation Research Part C*, vol. 107, pp. 287–300, 2019.

[19] D. Li, C. Zhang, and J. Cao, "Short-term passenger flow prediction of a passageway in a subway station using time space correlations between multi sites," *IEEE Access*, vol. 8, pp. 72471–72484, 2020.

[20] Z. Jinlei, C. Hongshu, C. Feng, M. Wei, and H. Zhengbing, "Short-term origin - destination demand prediction in urban rail transit systems: a channel-wise attentive split-convolutional neural network method," *Transportation Research Part C*, vol. 124, p. 102928, 2021.

[21] Z. Xu, R. Zhu, Q. Yang, L. Wang, R. Wang, and T. Li, *Short-Term Bus Passenger Flow Forecast Based on the Multi-Feature Gradient Boosting Decision Tree*, Springer, 2020.

[22] Y. Liu, X. Luo, and M. Yang, "Research on passenger flow prediction of bus line based on gradient boosting decision tree," in *2020 Chinese Control And Decision Conference (CCDC)*, Hefei, China, August 2020.

[23] Z. Zhang, C. Wang, Y. Gao, J. Chen, and Y. Zhang, "Short-term passenger flow forecast of rail transit station based on MIC feature selection and ST-LightGBM considering transfer passenger flow," *Scientific Programming*, vol. 2020, Article ID 3180628, 15 pages, 2020.

[24] L. Liu, R.-C. Chen, Q. Zhao, and S. Zhu, "Applying a multi-stage of input feature combination to random forest for

improving MRT passenger flow prediction," *Computing*, vol. 10, no. 11, pp. 4515–4532, 2019.

[25] W. Zhou, W. Wang, and D. Zhao, "Passenger flow forecasting in metro transfer station based on the combination of singular spectrum analysis and AdaBoost-weighted extreme learning machine," *Sensors*, vol. 20, no. 12, p. 3555, 2020.

[26] S. Lin and H. Tian, "Short-term metro passenger flow prediction based on random forest and LSTM," in *2020 IEEE 4th information technology, networking, Electronic and Automation Control Conference (ITNEC)*, pp. 2520–2526, Chongqing, China, June 2020.

[27] X. Dong, T. Lei, S. Jin, and Z. Hou, "Short-term traffic flow prediction based on XGBoost," in *2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS)*, pp. 854–859, Enshi, China, May 2018.

[28] Q. Du, F. Yin, and Z. Li, "Base station traffic prediction using XGBoost-LSTM with feature enhancement," *IET Networks*, vol. 9, no. 1, pp. 29–37, 2020.

[29] Y. Jing, H. Hu, S. Guo, X. Wang, and F. Chen, "Short-term prediction of urban rail transit passenger flow in external passenger transport hub based on LSTM-LGB-DRS," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4611–4621, 2020.

[30] X. Wang, L. Huang, H. Huang, B. Li, Z. Xia, and J. Li, "An ensemble learning model for short-term passenger flow prediction," *Complexity*, vol. 2020, Article ID 6694186, 13 pages, 2020.

[31] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, San Francisco California USA, August 2016.