Hindawi
Journal of Advanced Transportation
Volume 2021, Article ID 5559562, 24 pages
https://doi.org/10.1155/2021/5559562

Research Article

Spatio-Temporal Segmented Traffic Flow Prediction with ANPRS Data Based on Improved XGBoost

Bo Sun[1,2] **Tuo Sun**[1,3] **and Pengpeng Jiao**[1]

[1]*Beijing Key Laboratory of General Aviation Technology, Beijing University of Civil Engineering and Architecture, Beijing 100044, China*
[2]*Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China*
[3]*Key Laboratory of Road and Traffic Engineering of the Ministry of Education, Tongji University, Shanghai 201804, China*

Correspondence should be addressed to Pengpeng Jiao; jiaopengpeng@bucea.edu.cn

Received 16 February 2021; Revised 22 April 2021; Accepted 19 May 2021; Published 31 May 2021

Academic Editor: Jinjun Tang

Traffic prediction is highly significant for intelligent traffic systems and traffic management. eXtreme Gradient Boosting (XGBoost), a scalable tree lifting algorithm, is proposed and improved to predict more high-resolution traffic state by utilizing origin-destination (OD) relationship of segment flow data between upstream and downstream on the highway. In order to achieve fine prediction, a generalized extended-segment data acquirement mode is added by incorporating information of Automatic Number Plate Recognition System (ANPRS) from exits and entrances of toll stations and acquired by mathematical OD calculation indirectly without cameras. Abnormal data preprocessing and spatio-temporal relationship matching are conducted to ensure the effectiveness of prediction. Pearson analysis of spatial correlation is performed to find the relevance between adjacent roads, and the relative importance of input modes can be verified by spatial lag input and ordinary input. Two improved models, independent XGBoost (XGBoost-I) with individual adjustment parameters of different sections and static XGBoost (XGBoost-S) with overall adjustment of parameters, are conducted and combined with temporal relevant intervals and spatial staggered sectional lag. The early_stopping_rounds adjustment mechanism (EAM) is introduced to improve the effect of the XGBoost model. The prediction accuracy of XGBoost-I-lag is generally higher than XGBoost-I, XGBoost-S-lag, XGBoost-S, and other baseline methods for short-term and long-term multistep ahead. Additionally, the accuracy of the XGBoost-I-lag is evaluated well in nonrecurrent conditions and missing cases with considerable running time. The experiment results indicate that the proposed framework is convincing, satisfactory, and computationally reasonable.

1. Introduction

As a key technological component of intelligent transportation systems (ITS), traffic flow prediction has become an extensively researched topic. To support the dynamic application of ITS, traffic forecasting models usually predict traffic fluctuation, ranging from seconds to hours [1]. Traffic prediction is proved to act a significant role in providing more accurate online traffic demand prediction for traffic control, management, and guidance [2]. In recent years, with the available spatio-temporal availability of various detectors and the advanced intelligent computing, traffic flow prediction is extended to the network range and data-driven condition with nonrecurrent cases [3]. The traffic flow can be affected by nonrecurrent events such as road construction, sport events, and weather changes, which will lead to spatio-temporal deviations of traffic patterns, compared with regular cases. Meanwhile, the upstream flow and downstream flow show obvious spatial relevance of traffic propagation. To capture both temporal and spatial relationships of the traffic network, the cooperativity and generalization of traffic flow prediction should be improved. How to predict the traffic flow quickly with consideration of spatial cooperativity by congestion propagation of segment upstream and downstream flow and temporal multiple-step prediction, fully consider nonrecurrent generalization, and improve computing efficiency by time horizon of related timesteps remain to be investigated and answered by this paper.

Based on the issues above, traffic prediction problems particularly about highway segments have attracted a lot of attention as the rapid emerging and closing connection of cities. The accurate prediction of highway has obvious influence on logistics, trade, and commuting. At present, there are a number of sensors and cameras to help us obtain traffic data, such as Automatic Number Plate Recognition System (ANPRS), which is installed along mainlines and exits and entrances of toll stations in the highway, is a popular form of expert system that has been applied in many countries. This system is much needed for the detection of vehicles and to optimize all functions, including monitoring, controlling, problem solving, fine management, and compliance. Based on the ANPRS, travel time, traffic volume, travel route, and other traffic data can be acquired. Current historical research utilizes historical ANPRS data on mainlines to predict traffic state and traffic congestion, such as iterative tensor decomposition (ITD) [4], order-k Markov model [5], K-nearest neighbors (KNN) [6], dynamic linear model (DLM) [7], and linear regression [8]. However, these ANPRS datasets on the mainline can only receive traffic data and be studied in single direction. As we all know, the general segments are usually divided into two directions. At the same time, it is not accurate enough to only consider the traffic prediction on the mainline segment. Above methods usually ignore information about exits and entrances of mainline and predict directly with the ANPRS datasets with sparse distance. For instance, if there are exits and entrances of service area, interchange, and ramps in the toll stations, usually there will be four directions that remain to be detected. A segment of a highway usually has a toll station in each direction, and each toll station has a set of entrance and exit. Ignoring the traffic information might greatly reduce the accuracy of prediction for weaving sections in the segments and affect the performance of traffic control, management, and guidance. How to predict the traffic flow if the highway with consideration of high-resolution sections' division mode based on ANPRS is worthwhile to be resolved.

Plenty of machine learning (ML) models are used to predict the traffic flow. Existing ML methods are still full of challenges for how to deal with big data [9]. It is still worth discussing and studying how to further improve the prediction accuracy of highway and capture spatio-temporal information and data analysis [10, 11]. In addition, by acquiring the vast traffic sensing data, real-time traffic flow prediction is becoming an important part of traffic control, accident reporting, and intelligent transportation systems [12]. Many machine learning methods have been used in the field of traffic prediction. Since eXtreme Gradient Boosting (XGBoost) was proposed in 2014, it has been favored by a number of scholars. However, XGBoost in the transportation fields has not yet been more developed and applied. The extreme parallel optimization of the XGBoost method not only reduces overfitting but also reduces computing time. XGBoost controls the complexity of problems and can greatly improve the efficiency of the algorithm.

At present, a large number of studies [13] about the highway problems are only based on the location of detectors to obtain traffic data. But in the highway, due to the high cost, the distance between adjacent detectors is often far, so it is impossible to obtain more detailed traffic state directly. It is necessary to introduce a generalized extended-segment data acquirement mode for the highway to accurately predict and find out the source of congestion. We divide the segment into different sections according to whether there are cameras to capture traffic information directly. The main contributions of this paper are summarized as follows.

First, the segment data can be obtained directly when there are cameras in the range of the road section. Second, based on ANPRS, we propose a section-flow calculation method for the highway to predict the traffic state finely and microcosmically. For the segment flow which cannot be obtained directly, we take the OD relationship between entrances and exits of toll stations and the license plate recognition relationship of upstream and downstream roads for mathematical calculation. The established calculation method about highway can handle all similar cases and can be extended to the same scenarios for data acquisition and road section division to further manage and prevent traffic congestion. Third, compared with other prediction methods, XGBoost has advantages in scalability, high efficiency, low calculation cost, supporting for parallelization, and regularization processing. Here, we propose an improved XGBoost-based spatio-temporal method with the EAM optimization mode to predict the traffic flow of the segmented highway, by considering of multiple-step short-term and long-term prediction, influence of nonrecurrent incidents, and spatial interaction of sophisticated staggered sections.

The paper is organized according to the following parts. The next section summarizes related literature review. Section 3 introduces the basic methodology framework and formation mechanism of the XGBoost model. In Section 4, data acquirement as well as analysis and processing are described in detail, where the training set and testing set are specifically divided. In Section 5, parameter adjustments of XGBoost models, special events, and missing cases are discussed, and the accuracy of SARIMA, CNN, RF, and LSTM are compared with variants of XGBoost by considering spatial lag. Finally, a conclusion and future research plan are given in the last section.

2. Literature Review

Traffic prediction is mainly divided into parametric and nonparametric methods [14]. Parametric methods mainly include autoregressive integrated moving average model (ARIMA) [15], Box–Jenkins time series model [16], linear regression (LR) [17], Kalman filter (KF) [18], random forest (RF) [19], exponential smoothing (ES) [20], and fuzzy C-means (FCM) [11]. Many scholars have proposed variants of ARIMA to improve prediction accuracy, such as ARIMAX, which is made up of explanatory variables [21] and seasonal autoregressive integral moving average (SARIMA) [22, 23]. Afterwards, more attention is drawn to spatial networks, temporal adaptive schemes, and some hybrid models.

Spatio-temporal ARIMA [24] is proposed to incorporate spatial influence. Huang et al. developed a combined model of ARIMA-ANN, in which the linear component is processed by ARIMA, while ANN deals with the nonlinear aspect. Consequently, the hybrid model fully improves prediction accuracy [25]. The ability of these models to grasp spatial dynamic and nonlinear characteristics is limited.

Due to the uncertainty of the traffic data structure and nonlinear relationship hidden behind datasets, nonparametric methods are more flexible and complex enough for the nonlinear relationship. Statistical methods, such as support vector machine (SVM), have been applied to predict the traffic flow [26, 27]; however, due to its sensitivity in selecting kernel functions and parameters, scholars obtain chaotic wavelet SVM, least squares SVM, particle swarm optimization SVM, and genetic algorithm SVM for optimization. KNN has been widely studied as well [28, 29]. Actually, these methods are still hard to deal with large-scale data problems. With the development of computation intelligence, the neural network (NN) is widely used in multidimensional and complex nonlinear prediction problems [30, 31]. Recently, with the emerging of deep learning, more deep and efficient structures are derived from NN, such as deep belief network (DBN) [32], fuzzy neural network (FNN) [33], convolutional neural network (CNN), and recurrent neural network (RNN). By memorizing the characteristics of temporal correlations, long short-term memory network (LSTM) [34, 35] and Bi-LSTM [36] have been proved to outperform RNN in traffic prediction. To utilize the superiority of different methods, hybrid methods are proposed to solve complex problems [37], such as highway and network prediction. For example, Ma et al. established a large-scale congestion prediction model based on a RNN and a restricted Boltzmann machine [38]. In fact, these deep learning methods are continuously optimized and are not easy to be trained because the structures and hyperparameters have many kinds in different cases. When it comes to nonrecurrent datasets, the overfitting problem is still tough, which leads to distinct strategies, such as hybrid prediction methods and dropout layer optimization and regularization.

Recently, XGBoost, which is a successful prediction method, has been applied in lots of issues of Kaggle competition and other applications with excellent results, such as Didi products. It is a decision tree-based method developed by Chen and Guestrin [39] and improved from Gradient Boosting Decision Tree (GBDT), which is a type of boosting algorithm [40]. GBDT generally passes multiple iterations, where each iteration produces a weak classifier, and each classifier is trained based on the residuals of the previous classifier. At present, GBDT was proposed to make short-term traffic prediction which was viewed as combining the strengths of boosting algorithms and decision trees [41]. However, XGBoost uses a gradient descent algorithm to optimize the differential loss function in order to generate a boosted set of weak prediction models [42]. In addition, compared to the traditional GBDT and statistical learning and ML methods, XGBoost employs a regularization strategy to control model complexity and greatly avoid overfitting. Furthermore, it holds efficient computing power, scalability, and less memory consumption [43]. In our study, based on the EAM mode, the proposed improved XGBoost models are superior to other methods in terms of prediction accuracy.

3. Methodology

3.1. Methodology Framework. Figure 1 shows the proposed methodology framework. Data part is composed of segment data obtained directly by cameras which can get detected traffic information and ANPRS data based on the OD relationship. The whole data part consists of data collection, calculation, preprocess, and spatio-temporal correlation analysis. The model part comprises adjusting multiple parameters, tree structure operation, model optimization of EAM mode, and results' evaluation.

3.2. XGBoost. XGBoost is based on the GBDT model and improves on the calculation speed of the algorithm, while optimizing its performance and efficiency, attempting to achieve the ultimate balance. Compared with GBDT, XGBoost explicitly adds the complexity of the tree structure as a regular term and uses second derivative information in the derivation of the optimization objective equation, whereas GBDT only uses the first-order allowance. XGBoost implements an approximate algorithm for the split-node search, which is used to quicken and reduce memory consumption. The node splitting algorithm automatically utilizes the feature of sparseness, and the data is sorted in advance and stored in the form of blocks, which are conducive to parallel computing.

The core idea of XGBoost [39] is that it continuously adds new trees and performs feature splitting to grow a tree during implementation. Each time a tree is added, it learns a new function to fit the pseudoresiduals of the last prediction. When we get K trees after training, we need to predict the score of a sample. In fact, according to the characteristics of this sample, each tree will fall into a corresponding leaf node. Accordingly, the scores must be summed corresponding to each tree which will be as the forecasting value of the sample.

XGBoost [44], as a tree integration model, sums the results of K trees, where $\widehat{y}_i$ is the final predicted value:

$$\widehat{y}_i = \phi(x_i) = \varepsilon \sum_{k=1}^{K} f_k(x_i), \quad f_k \in F. \tag{1}$$

Here, K represents the number of trees, f_k is the model of the k tree, and ε is the learning rate. Supposing that there are n samples and m features in a given sample set,

$$D = \{(x_i, y_i)\}, \quad (|D| = n,\ x_i \in R^m, \text{ and } y_i \in R), \tag{2}$$

where x_i represents the i sample and y_i represents the i category label, and the space F of the regression tree is

$$F = \left\{ f(x) = w_{q(x)} \right\} \quad \left(q\colon R^m \longrightarrow T \text{ and } w \in R^T\right). \tag{3}$$

Here, q represents the structure of each tree and maps samples to corresponding leaf nodes, T represents the

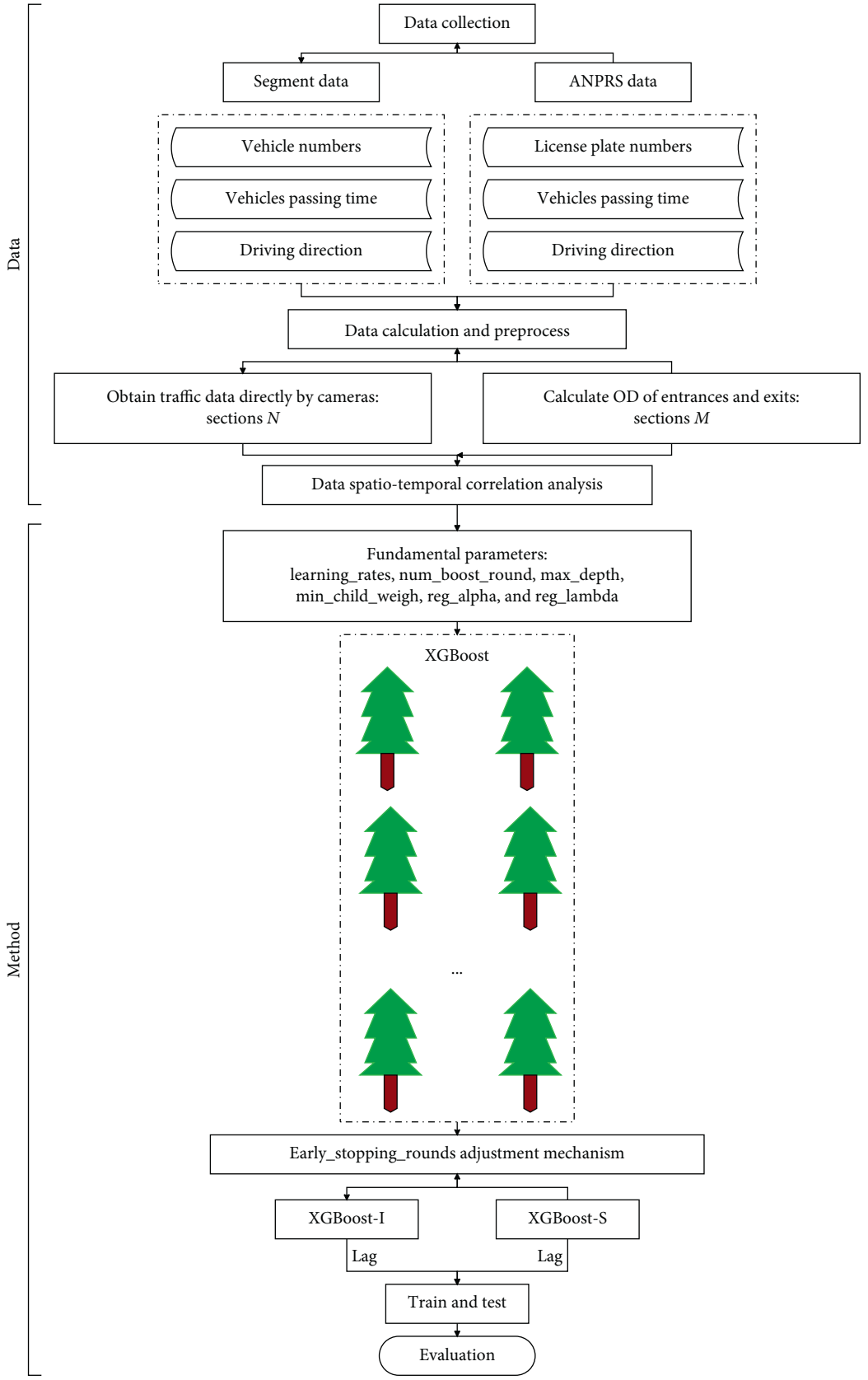


Figure 1: Methodology framework.

number of leaf nodes of each tree, w represents a set of leaf node scores of each tree, and $f(x)$ corresponds to the structure q of the tree and the weight w of the leaf nodes. Therefore, the predicted value of XGBoost is the sum of the values of the leaf nodes corresponding to each tree. In this study, the goal is to optimize K trees; hence, we minimize the following objective equation with a regular term to make the tree f_k suitable for training data under the max_depth constraint:

$$L(\phi) = \sum_i l(\hat{y}_i, y_i) + \sum_k \Omega(f_k). \tag{4}$$

In equation (4), the first term is training loss error and l is a differentiable convex loss function that measures the difference between predicted value $\hat{y}_i$ and target value y_i. The second term is a regular term, which controls the complexity of the tree and prevents overfitting. $\hat{y}_i$ is updated by adding a new tree weighted by the learning rate ε, which is represented by equation (5). Among them, γ and λ are regularization parameters which are used to adjust complexity of the tree:

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2. \tag{5}$$

r_i indicates that the residual term after the $i$thfitting is expressed by equation (6). Each time a new tree is added, the pseudoresidual of the last prediction must be fitted:

$$r_{i+1} = r_i - \hat{y}_i. \tag{6}$$

During training, a new f function is added in the new round to minimize the objective function. In the tth round, our objective equation is

$$L^{(t)} = \sum_{i=1}^{n} l\left(y_i, \widehat{y_i^{t-1}} + f_t(x_i)\right) + \Omega(f_t). \tag{7}$$

Next, we expand the objective function by Taylor expansion, taking the first three terms and removing the high-order small infinitesimal term. Finally, our objective function is transformed into equation (8), and g_i and h_i are the first derivative and second derivative, respectively:

$$L^{(t)} \approx \sum_{i=1}^{n}\left[l\left(y_i, \widehat{y^{t-1}}\right) + g_i f_t(x_i) + \frac{1}{2}h_i f_t^2(x_i)\right] + \Omega(f_t), \tag{8}$$

$$g_i = \partial_{\hat{y}^{(t-1)}} l\left(y_i, \widehat{y^{t-1}}\right), \tag{9}$$

$$h_i = \partial^2_{\hat{y}^{(t-1)}} l\left(y_i, \widehat{y^{t-1}}\right). \tag{10}$$

According to equations (3), (5), and (8),

$$\begin{aligned} L^{(t)} &\approx \sum_{i=1}^{n}\left[g_i w_{q(x_i)} + \frac{1}{2}h_i w^2_{q(x_i)}\right] + \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \\ &= \sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)w_j^2\right] + \gamma T. \end{aligned} \tag{11}$$

$I_j = \{i|q(x_i) = j\}$ is defined as the sample set of leaf j. When $q(x)$ is fixed, we transform the iteration about the tree model into an iteration about the leaf nodes of the tree. Accordingly, score w is found to correspond to the optimal leaf node j. The optimal value of the leaf node is brought into the objective function, and the corresponding value of the final objective function can be expressed as follows:

$$\begin{aligned} w_j &= -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}, \\ \bar{L}^{(t)}(q) &= -\frac{1}{2}\sum_{j=1}^{T}\frac{\left(\sum_{i\in I_j} g_i\right)^2}{\sum_{i\in I_j} h_i + \lambda} + \lambda T. \end{aligned} \tag{12}$$

In general, we cannot enumerate all possible tree structures and choose the optimal one; hence, we use a greedy algorithm as it can greatly improve computing efficiency. We start with a single leaf node and iteratively split it to add nodes to the tree. By enumerating the feasible segmentation points and selecting the minimum target function and maximum gain partition, the gain equation becomes

$$\text{Gain} = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma. \tag{13}$$

The above equation is used to evaluate the loss function after slicing. In practice, however, it is used to evaluate the candidate after slicing. The XGBoost model produces many simple trees that are used to assess scores of leaf nodes during splitting. The first, second, and third terms of the equations represent the scores on the left, right, and original leaves, respectively. In addition, γ is a regularization parameter on other leaves and is used during training. XGBoost supports parallelism. During the learning process, the features must be sorted by a loss function to determine the best segmentation point. To acquire model best performance, the appropriate and reasonable parameters in the XGBoost model must be set and adjusted for different tasks. Generally, the XGBoost model optimizes parameter settings by cross validation, which are described in Section 5.

4. Data

4.1. Data Sources. Our data is derived from Shaoxing, Zhejiang Province, China, as shown in Figure 2(a), and road network exhibition comes from Python package OSMnx [45]. The data range is from September 1st to November 19th, 2019 (23040 intervals over 80 days in total). The latitude range and longitude range of the target network are (120.523, 120.916) and (30.047, 30.153), and the total length of the target highway is about 39.25 kilometers, as shown in Figure 2(b), with a speed limit of 100 km/h. From southeast to northwest, the direction of the traffic flow is up-direction. However, from northwest to southeast, the direction of the traffic flow is down-direction. In total, there are three toll

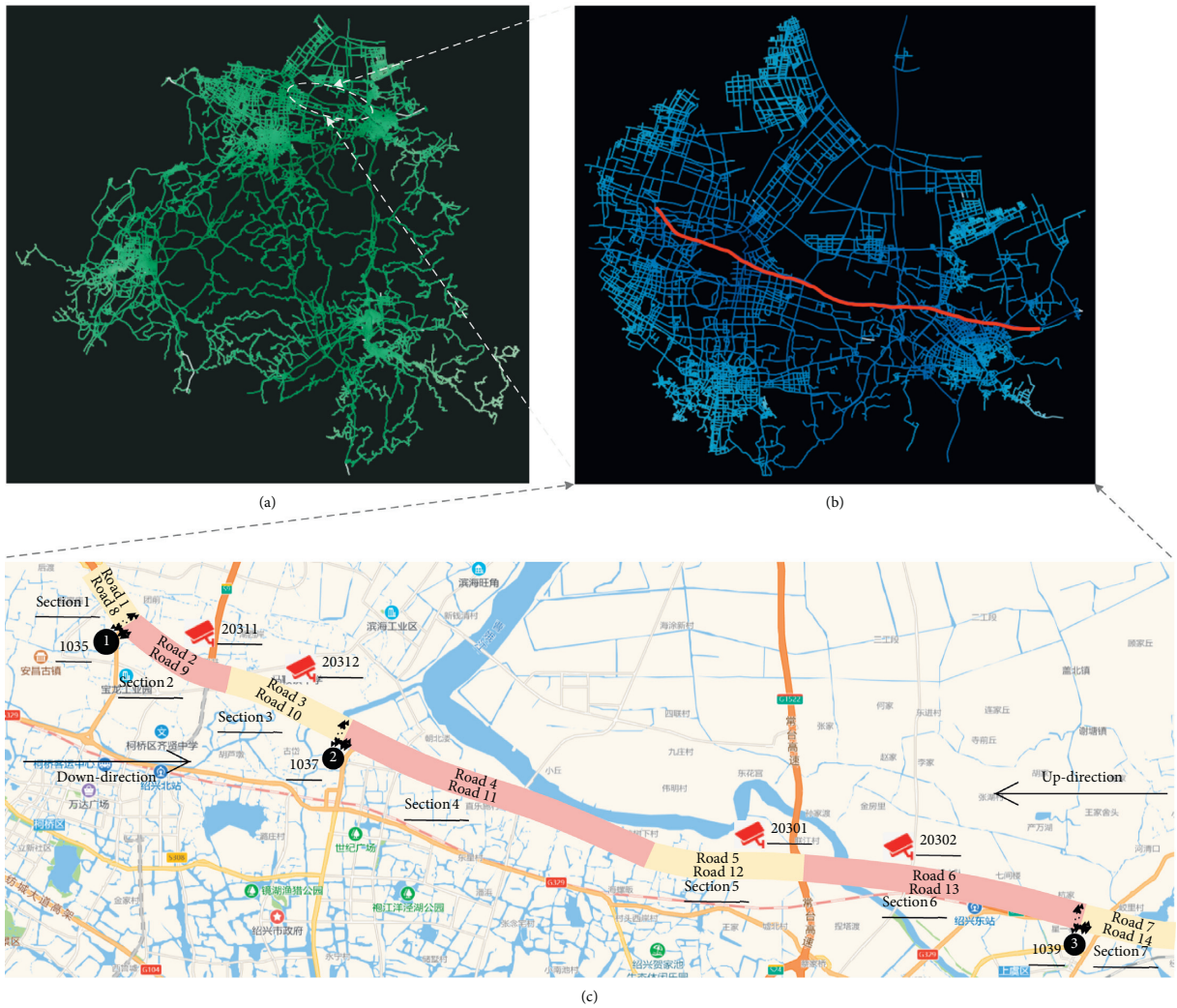


Figure 2: Target highway.

stations, as shown in Figure 2(c), which are 1035 Keqiao Toll Station (indicated by black circle 1), 1037 Shaoxing Toll Station (indicated by black circle 2), and 1039 Shangyu Toll Station (indicated by black circle 3), and every toll station has its own entrance and exit connected with up-direction and down-direction, that is, there are double entrance-and-exit combination. Moreover, four detected cameras are numbered officially 20311, 20312, 20301, and 20302 from northwest to southeast. Each camera can acquire real-time data including vehicle ID, passing time, passing site ID (toll station or detected camera), license plate numbers, and driving direction.

We divide the entire segment into 7 sections from northwest to southeast, which are labeled as Sections 1–7. Among them, Sections 1, 4, and 7 are virtually partial sections without any cameras, which do not have cameras to capture vehicle information and usually get different traffic information from adjacent sections, as entrances and exits of toll stations affect the flow of the mainline. A number of loop detectors and highway research data measure the traffic counts in each time interval, but the observation of the traffic flow does not differentiate road directions. In our study, based on the characteristics of up-direction and down-direction flow, each section is further divided into two roads. Up-direction from northwest to southeast is set as Road 1 to Road 7. The down-direction from northwest to southeast is set as Road 8 to Road 14. To illustrate in detail, Section 1 includes up-direction Road 1 and down-direction Road 8, while Section 2 includes up-direction Road 2 and down-direction Road 9 in Figure 2(c).

4.2. ANPRS of Generalized Segmented Data Acquirement. First, according to up-direction or down-direction, the vehicle is, respectively, mapped in different roads. Upstream and downstream traffic data of the respective ANPRS are counted according to license plate matching. The numbers of vehicles of every 5 minutes in the chronological order are also counted as segment data. It means that one license plate number corresponds to one vehicle. For toll stations, when vehicles pass entrances or exits, its license plate numbers can be selected and recorded by cameras. In the study of highway, there are four cameras. If we can only obtain the traffic data of four sections according to the location of

cameras directly, it is not conducive to better understand the traffic situations and OD law with toll stations and ramps. Therefore, the ANPRS data acquirement mode not only divides the whole segment more finely but also grasps more traffic information so as to accurately predict the future segmented traffic flow. The method is applicable to any highways with the same structure, such as service areas and roundabouts. In addition, these kinds of toll stations and the replacements including entrances and exits in the highway are universal in any country.. According to the above, we propose a generalized method for accurately acquiring data on highway segment.

To illustrate, for different sections, there are two ways to obtain the accurate traffic flow. The first way is directly obtain segment data through the cameras capturing the number of vehicles of the corresponding sections. In our target highway, Sections 2, 3, 5, and 6 can acquire segment data directly. The second way is that the corresponding sections are not clearly displayed in the traffic data such as Sections 1, 4, and 7. Hence, the data is obtained by performing the proposed calculation mode according to license plate recognition and OD flow relationship as the entrances and exits of each toll station.

The generalized extended-segment data calculation mode is as follows. According to Figure 3, we calculate the up-direction traffic flow (Road A1) and down-direction traffic flow (Road B1) of Section X1, as shown in the following equations:

$$f_{\text{Down}}^{\text{Road }B1} - f_{\text{Down-exit}}^{S1} + f_{\text{Down-entrance}}^{S1} = f_{\text{Down}}^{R1}, \tag{14}$$

$$f_{\text{Down-entrance}}^{S1} + f_{\text{Up-entrance}}^{S1} = f_{\text{entrance}}^{S1}, \tag{15}$$

$$f_{\text{Up}}^{R1} - f_{\text{Up-exit}}^{S1} + f_{\text{Up-entrance}}^{S1} = f_{\text{Up}}^{\text{Road }A1}, \tag{16}$$

$$f_{\text{Up-exit}}^{S1} + f_{\text{Down-exit}}^{S1} = f_{\text{exit}}^{S1}. \tag{17}$$

After vehicles enter through the descending entrance of the toll station $S1$, the license plate numbers are detected at the toll station and continued to be captured by the camera $R1$. Through license plate recognition and comparison, the number of vehicles with the same license plate numbers is $f_{\text{Down-entrance}}^{S1}$. The total number of vehicles at the toll station $S1$ entrance will be obtained by considering the previous data processing stage. According to equation (16), $f_{\text{Up-entrance}}^{S1}$ is also known.

Part of the vehicles pass through the camera $R1$, pass up-direction exit of toll station $S1$, and leave the highway, where the license plate numbers are detected at the toll station. These vehicles are then first captured by the camera $R1$. By comparing the license plate, the number of vehicles with the same license plate numbers can be expressed as $f_{\text{Up-exit}}^{S1}$. The total number of vehicles at toll station exit $S1$ will be obtained by analyzing the previous data. And, according to equation (18), $f_{\text{Down-exit}}^{S1}$ is also known. So far, other than the up-direction Road $A1$ and down-direction Road $B1$ of Section X1, which are unknown, the remainders are all known. Therefore, the flow of Road $A1$ and Road $B1$ can be obtained by the following equation:

$$\left(\begin{array}{l} f_{\text{Down}}^{\text{Road }B1} = f_{\text{Down}}^{R1} - f_{\text{Down-entrance}}^{S1} + f_{\text{exit}}^{S1} - f_{\text{Up-exit}}^{S1}, \\ f_{\text{Up}}^{\text{Road }A1} = f_{\text{Up}}^{R1} - f_{\text{Up-exit}}^{S1} + f_{\text{entrance}}^{S1} - f_{\text{Down-entrance}}^{S1}. \end{array} \right). \tag{18}$$

For our target highway, traffic flow data of Section 1 (Road 1 and Road 8), Section 4 (Road 4 and Road 11), and Section 7 (Road 7 and Road 14) can be obtained by performing the same calculation. The flow of all sections (Road 1–Road 14) can be obtained as ground truth, laying the foundation for subsequent traffic prediction studies.

4.3. Data Preprocess and Hardware. The outliers in the datasets will far exceed the ground truth and greatly affect the accuracy of prediction. In order to suppress the influence of outliers, we apply winsorization to preprocess data [46]. Winsorization attempts to replace the minimum and maximum values within a dataset with their closest values. Winsorization is especially useful when dealing with traffic data influenced by incidents and occasional factors such as adverse weather or traffic accidents. Since no event and weather records are available for the traffic datasets used in this study, winsorization plays a vital role in suppressing the effect of extreme values. Mathematically, winsorization is represented by equation (19). Assuming that the value of the sequence to be processed is given by w, where $w = (w_1, w_2, \ldots, w_k)$, the processed value w_i^K following winsorization will be

$$w_i^K = \left\{ \begin{array}{ll} w_{i+1}, & \text{if } w_i = \min(w), \\ w_i, & \text{if } \min(w) < w_i < \max(w), \\ w_{k-1}, & \text{if } w_i = \max(w). \end{array} \right\}. \tag{19}$$

Here, w_i is the i pending value, $\min(w)$ is the minimum value of the pending value, $\max(w)$ is the maximum value of the pending value, k is the number of pending values, and w_i^K is the winsorized value of the i pending value.

We divide the datasets of 80 days into 75 days for the training set and 5 days for the testing set. Moreover, numerous training iterative processing is conducted to find the recursive relationship in the traffic flow in order to attain more accurate prediction. The entire datasets take the past six 5 min flow data so as to predict the future. Python library Keras, which is based on Tensorflow, is used to build our models. All experiments are performed by a PC Server with the following configuration: Intel(R), Xeon(R), CPU E5-1650, 3.50 GHz, and 64 GB of memory.

4.4. Exploring Spatio-Temporal Correlations for Lag Calibration. There is a spatial transmitting correlation between the highway traffic flow of different sections. In order to prove it, we use the Pearson correlation test given in the following equation to test that T in the datasets and the next section lag by intervals of one 5 min, two 5 min, three 5 min,

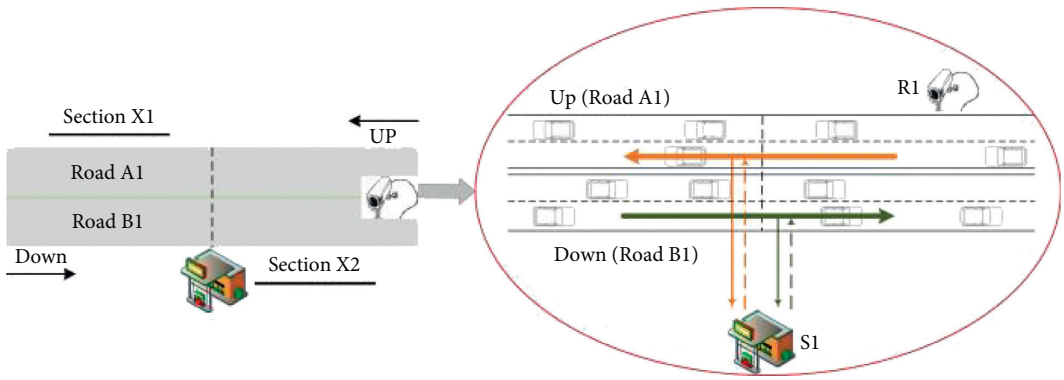


Figure 3: Schematic diagram of traffic flow calculation for Section X1.

and four 5 min for spatial correlation between spatial variables:

$$\mathrm{Corr}(Y,Z) = \frac{E\left[(Y - E(Y))'(Z - E(Z))\right]}{E\left[(Y - E(Y))^2\right]E\left[(Z - E(Z))^2\right]}, \quad (20)$$

where Y and Z are two random variables with the same number of observations. In our study, Y and Z represent the data correlation between the front road in the same direction and the adjacent following road in Figure 2. The adjacent following road is the data of the front road lagging T intervals.

We calculate T intervals without lag and then calculate $T + 1$, $T + 2$, $T + 3$, and $T + 4$ of the lagging intervals of the next section of the adjacent section. As shown in Figure 4, the left picture is up-direction Road 1 to Road 7 and the right picture is down-direction Road 8 to Road 14. The best correlation performance occurs at $T + 1$, which is better than T without lag. As the spatial distance continues to increase, the average correlation gradually decreases sharply. It indicates that there is a strong spatial correlation between each road and its adjacent section. On the contrary, it is not surprising that a variable with a short lag T interval, which has a high correlation, but a variable with a large lagging interval, also has a certain degree of correlation with the previous section. This correlation analysis of the datasets provides evidence for setting spatial lag with the length of one interval which is really necessary in order to predict accurately.

5. Case Studies

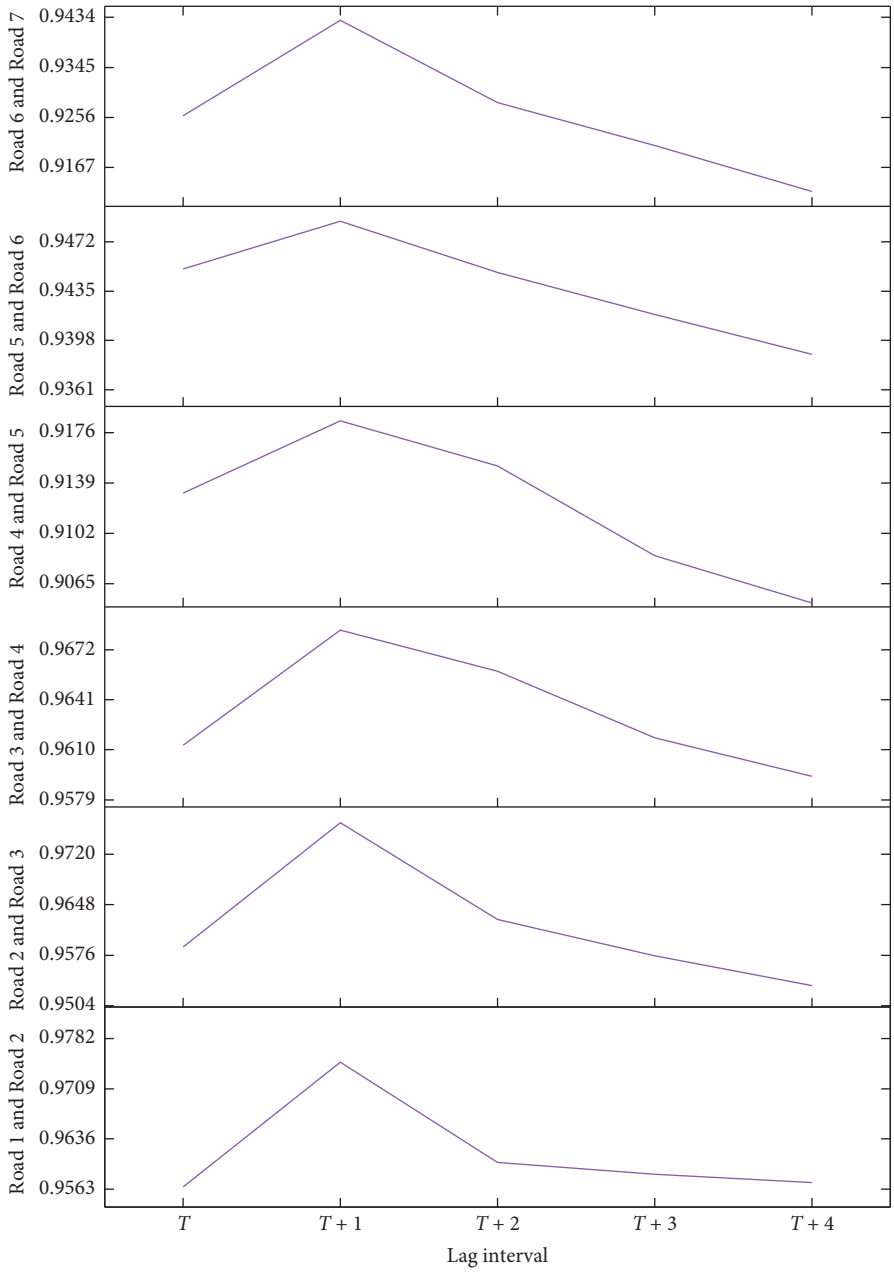
5.1. XGBoost Parameter Adjustment. We independently adjust the XGBoost parameters of the corresponding sections for each of the 7 sections of up-direction and down-direction, named XGBoost-I. When training iterations, early_stopping_rounds adjustment mechanism (EAM) for adjusting parameters is introduced to improve the XGBoost method. When the lowest error iteration comes up, the model continues to iterate 100 times. After that, if no lower error is found, the iteration is terminated. Otherwise, the model will repeat the above EAM mode. This is done to avoid missing optimal parameters until the best case.

Num_boost_round, which refers to the number of boosting trees, represents the number of training iterations. A value that is too small can result in underfitting, while a value that is too large can cause overfitting. Num_boost_round and learning_rates are generally adjusted with the same parameters, where learning_rates is comprised with a list of learning rates each time. The adjustment of up-direction and down-direction of the num_boost_round and learning_rates parameters of XGBoost-I is shown in Figure 5. We choose the minimum average Root Mean Square Error (RMSE) shown in equation (21) of 14 roads in up-direction or down-direction for all parameters including num_boost_round and learning_rates parameter settings. Here, learning_rates is 0.04, and the lowest mean RMSE is determined to be 24.1448. The num_boost_round results of the 14 roads (Roads 1–14) are shown in Table 1.

During model training, the other parameters also need to be determined. max_depth is the maximum depth of a tree by increasing the value to make the model more complex and avoid overfitting. Min_child_weight determines the minimum leaf nodes' sample weight, which is used to avoid overfitting as well. When the value is large, the model can avoid learning special local samples. We adjust max_depth and min_child_weigh synchronously, and the adjustments of up-direction and down-direction are shown in Figure 6. Both max_depth and min_child_weigh are traversed of the whole range from 1 to 10, and the corresponding best parameters are recorded as 3 and 10. The lowest mean RMSE is found to be 23.6388.

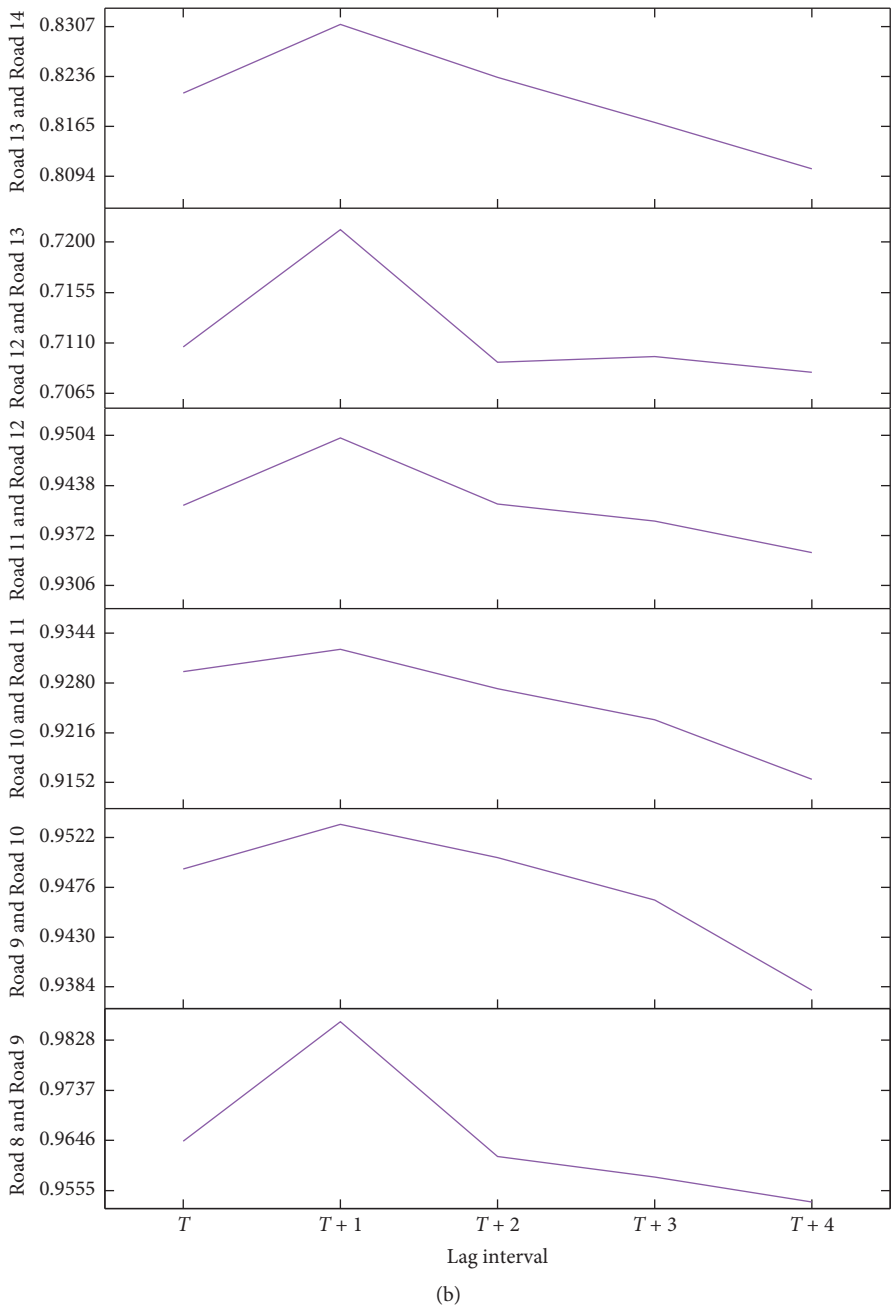
Reg_alpha is the $L1$ regularization term of the weight, which improves the processing speed of the model. Reg_lambda is the weighted $L2$ regularization and is used to control the fitting situation of XGBoost. We synchronize reg_alpha and reg_lambda, as shown in Figure 7, in terms of up-direction and down-direction adjustments. The best parameters corresponding to reg_alpha and reg_lambda are 0.05 and 0.1, respectively. The parameters have little effect on the error result, and the lowest mean RMSE is 23.6285.

Gamma specifies the minimum dropping loss function which is required for node splitting. Subsample is a ratio set that is used to train the model subsamples to the entire training process. This parameter controls the ratio of random sampling for each tree. Setting scale_pos_weight



(a)

Figure 4: Continued.



(b)

Figure 4: Pearson correlation results of up-direction and down-direction.

enables the algorithm to converge faster. Evals is a list that evaluates elements in the list during training, allowing to observe the effect of the validation set during training. Common parameters are used to control the macrofunction of XGBoost. The learning objective parameter is used to control the ideal optimization goal and measurement in the result of each step.

XGBoost consists of over thirty hyperparameters; hence, we choose the following parameters that confer greater impact on the optimization performance. The best relevant parameter settings of the XGBoost-I model are depicted in Table 2.

Here, we set another static XGBoost (XGBoost-S) model in regard to the overall adjustment of parameters among all sections of up-direction and down-direction. Specifically, the optimal situational parameters are adopted for the 14 roads, and its adjustment parameter settings are shown in Table 3.

5.2. Evaluation Index. For the evaluation of different prediction methods, we employ Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE) as the evaluation index. Given the

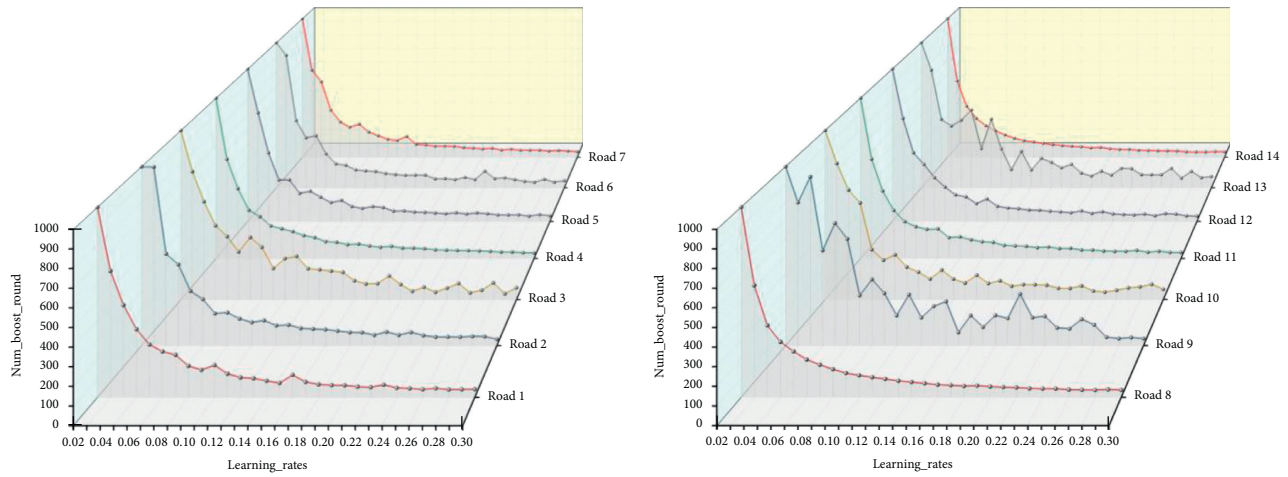


Figure 5: Num_boost_round and learning_rates up-direction and down-direction parameters' adjustment results.

Table 1: num_boost_round results of 14 roads.

Dir.	Section	num_boost_round	Dir.	Section	num_boost_round
Up	Road 1	358	Down	Road 8	219
	Road 2	453		Road 9	531
	Road 3	437		Road 10	571
	Road 4	301		Road 11	299
	Road 5	271		Road 12	373
	Road 6	344		Road 13	424
	Road 7	340		Road 14	280

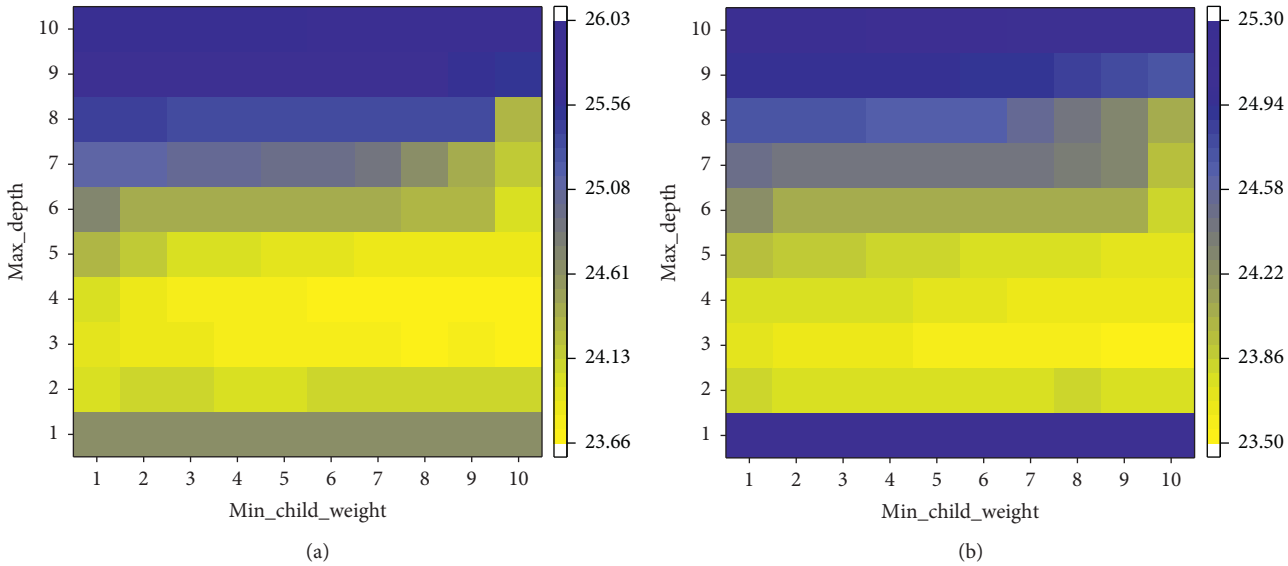


Figure 6: max_depth and min_child_weigh up-direction and down-direction parameter adjustment results.

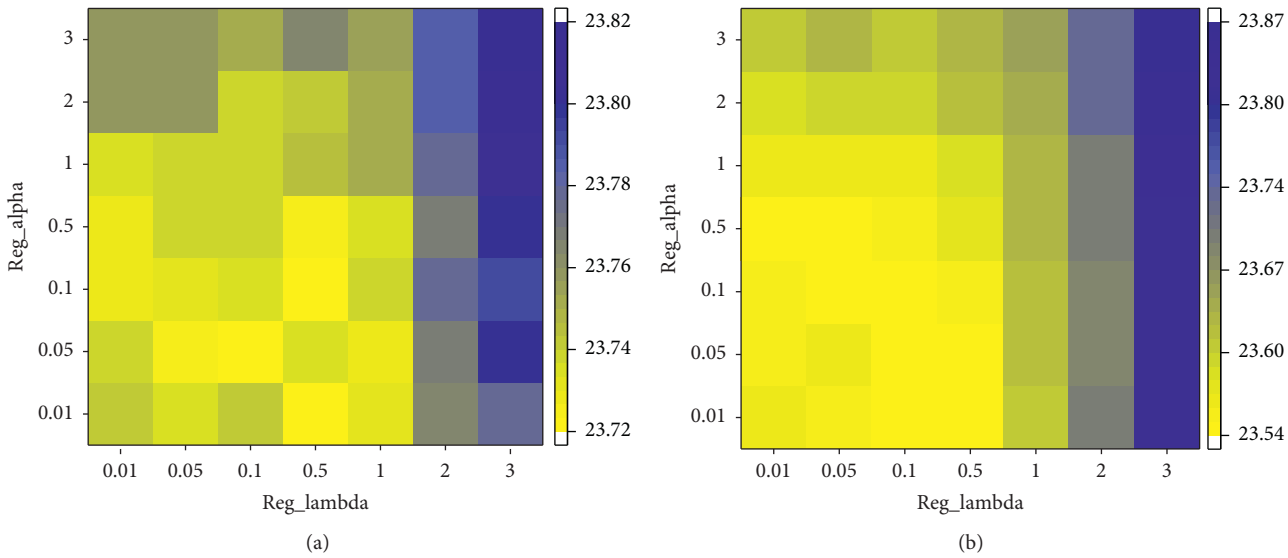


Figure 7: reg_alpha and reg_lambda up-direction and down-direction parameter adjustment results.

Table 2: Parameter settings of XGBoost-I

Type	Parameter	Settings
Booster	max_depth	3
	min_child_weight	10
	gamma	0
	subsample	1
	reg_alpha	0.1
	reg_lambda	0.05
	scale_pos_weight	1
General	Booster	gbtree
	Silent	0
	Nthread	max
Learning target	Objective	reg:gamma
	eval_metric	Depending on objective
	Seed	0
	learning_rates	0.04
	eval_metric	rmse
	evals	evallist

Table 3: Parameter settings of XGBoost-S.

Type	Parameter	Settings
Booster	max_depth	5
	min_child_weight	10
	gamma	0
	subsample	1
	reg_alpha	0.01
	reg_lambda	0.05
	scale_pos_weight	1
General	Booster	gbtree
	Silent	0
	Nthread	max
Learning target	Objective	reg:gamma
	eval_metric	Depending on objective
	Seed	0
	learning_rates	0.25
	num_boost_round	80

predicted value $\hat{f}_t$ and the ground truth f_t, RMSE, MAE, and MAPE are calculated as follows:

$$\text{RMSE} = \sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(f_t - \hat{f}_t\right)^2}, \tag{21}$$

$$\text{MAE} = \frac{1}{T}\sum_{t=1}^{T}\left|\left(f_t - \hat{f}_t\right)\right|, \tag{22}$$

$$\text{MAPE} = \sum_{t=1}^{T}\left|\frac{f_t - \hat{f}_t}{f_t}\right| \times \frac{100}{T}. \tag{23}$$

5.3. Spatial Lag. This model is divided according to the datasets in Section 4.3, which are trained and tested from two perspectives: temporal and spatial. Time series is signified in temporal data, whereas movement of the position is based on spatial data of seven sections used as input. Here, we explore two input methods: lag input and ordinary input. The time series is organized according to the past several 5 min, while the spatial position follows the traffic statistics of the seven sections (Roads 1–14) as input. For the entire target highway, when the vehicles travel from Section 1 to Section 7, or between any section, or dynamically from the current section to the next section of temporal and spatial displacement, the downstream traffic is transmitted by upstream, and the lagging relationship exists between upstream and downstream along with spatial lag. Therefore, the lag input is able to consider to reflect the propagation law of the traffic flow across the entire highway segment. As the traffic statistics' interval is 5 min, we take the spatial lag of one 5 min as input for each direction of upstream and downstream. Generally, driving speed will not exceed the speed limit, but vehicles on the highway generally are driven close to the speed limit. According to the calculation results

in regard to the speed limit and the Pearson spatio-temporal correlation test in Section 4.4, the rationality and effectiveness of one 5 min spatial lag is verified. Figure 8 shows the spatial lag of the target highway.

5.4. XGBoost-Related Models. For XGBoost models examined in this study (XGBoost-I and XGBoost-S are ordinary input without spatial lag), input modes are calibrated about different analyses above, called XGBoost-I-lag and XGBoost-S-lag, respectively, by spatial lag. Comparisons are made for highway traffic prediction of seven sections (14 roads). The according prediction results of up-direction and down-direction are given in Tables 4 and 5 . Before comparing with baseline methods, it is necessary to study the corresponding different methods of XGBoost.

On the whole, in regard to fourteen roads of up-direction and down-direction, except Roads 4, 7, and 14, RMSE and MAE of the XGBoost-I-lag model are found to be optimal. For MAPE, except for Roads 3, 4, 6, 7, and 13, XGBoost-I-lag is the best among the other 9 roads. Among the average results of all these roads, RMSE, MAE, and MAPE of the XGBoost-I-lag model are found to be better than those of the XGBoost-I model by 3.33%, 3.99%, and 2.87%, respectively. RMSE, MAE, and MAPE outperform those of the XGBoost-S model by 7.14%, 4.68%, and 5.97%, respectively, and are better than the XGBoost-S-lag model by 4.33%, 1.29%, and 2.38%, respectively. Notably, these three errors of the XGBoost-S-lag model are observed to be better than the XGBoost-S model by 3.02%, 3.56%, and 3.81%, respectively. Overall, the XGBoost-I-lag prediction result is considered to be the most accurate because it is due to the respective adjustment of different road parameters. Moreover, the fourteen roads, whose results are individually adjusted corresponding to the XGBoost-I parameter model, are also found to be better than the overall adjustment of the XGBoost-S parameter model. That is, the separate optimal parameter structure of each road is evidently better than the entire optimal parameter structure. In addition, the spatial lag input results of both XGBoost-I and XGBoost-S are better than the ordinary input. On the contrary, concerning different segment features of the fourteen roads, errors of Section 1 (Road 1 and Road 7), Section 4 (Road 4 and Road 11), and Section 7 (Road 7 and Road 14) are slightly larger than other sections. One possible reason is that the flow of these three sections is calculated using proposed formula deduction, and slight differences in the results are directly captured by the cameras. Therefore, improving the quality and maintenance of data acquisition equipment is still necessary for traffic prediction, and it is really necessary to expand segments for fine traffic information prediction. For Roads 4, 7, and 14, the best data interval should not be 5 minutes lag, so the XGBoost-I method of ordinary input is better than XGBoost-I-lag.

Figure 9 demonstrates the comparison between the predicted value and the ground truth of four types of XGBoost models on the up-direction of Road 2 that is predicted by the testing set.

The accuracy of the proposed XGBoost methods is verified using three types of errors in traffic prediction that are superimposed to various periods of twenty-four hours. Figure 10 depicts the performance of up-direction and down-direction traffic prediction using the proposed methods, respectively. In 24 h point-line plot, the distribution range of the prediction error of the XGBoost-I-lag model is shown with the solid red line representing average of errors. In addition, the black line represents average errors of XGBoost-I, the yellow line represents average errors of XGBoost-S, and the blue line represents average errors of XGBoost-S-lag. When checking the errors corresponding to the flow level, the prediction accuracy within prediction time is observed to continuously change with peak hours and nonpeak hours. When checking for errors based on the percentage deviations gathered from the observations, MAPE is used to judge the prediction accuracy. Evidently, with increase in the traffic flow, both RMSE and MAE rise substantially. When the observed flow is low, especially at late night and early morning, RMSE and MAE are lower because both they only consider the magnitude of deviation between the predicted value and observed value. Similarly, when examining the nature of the error corresponding to the time of day, compared to nonpeak hours, MAPE is seen to be relatively low during peak hours. To illustrate, the proposed model with the improved EAM mode totally performs well during the whole twenty-four hours. And, the effect of lag input is fully reflected.

Figure 10 infers that the XGBoost-I-lag model (solid red line) can provide accurate and stable traffic flow prediction. During peak hours, the model can be used to predict the traffic flow in a little failure. Therefore, reliable and accurate prediction of the traffic flow during times of heavy traffic is critical. Implementing alternative traffic management strategies as traffic managers can avoid traffic disruptions and provide decision-supporting solutions which should be conducted.

5.5. Special Case. Figures 11 and 12 show the prediction results for special days provided by the XGBoost-I-lag model. The blue line represents the predicted value, and the red line represents the ground truth of the traffic flow. Accordingly, the entire performance of the XGBoost-I-lag model is found to be really good in normal traffic conditions, while being really effective during special time. Two special traffic events, which are recorded during the day of the simulation, can be the reason why the traffic state is becoming congested from no congestion state. The XGBoost-I-lag model is able to capture the sudden change. A traffic accident causes continuing congestion for one hour, while occurring around 7:00. After the incident is handled, normal traffic operations are restored. Moreover, the weather event, which is abnormal (rain, snow, fog, etc.) long after 17:00, is ending at approximately 21:00. During the special period, severe traffic congestion and slow driving behaviors occur, and the traffic flow is greatly reduced as shown. In theory, the XGBoost-I-lag model is able to handle complex interactions of input variables and can make reasonable

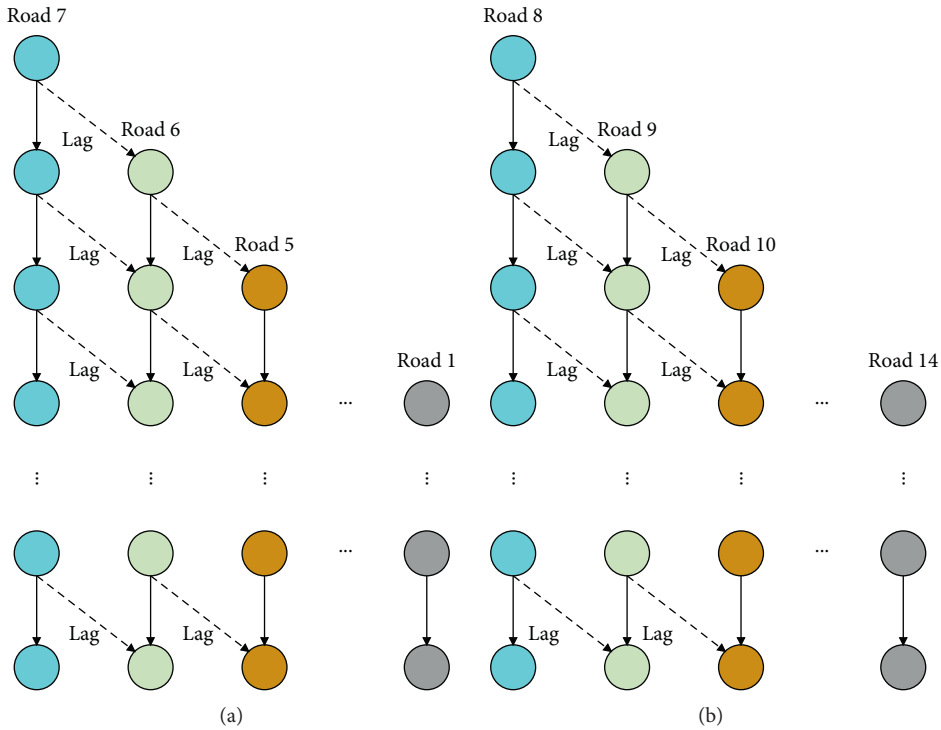


Figure 8: Spatial lag of up-direction and down-direction.

Table 4: Up-direction-related XGBoost models' prediction results.

Methods	Error	Road 1	Road 2	Road 3	Road 4	Road 5	Road 6	Road 7
XGBoost-I	RMSE	33.6726	17.5608	20.2317	**25.0574**	19.4625	20.8586	**29.1977**
	MAE	23.8850	13.0644	15.1233	**18.3011**	14.4374	14.0289	**19.9576**
	MAPE	13.9770	12.1249	13.7570	**10.8040**	12.2136	**12.0916**	**14.6106**
XGBoost-I-lag	RMSE	**31.8362**	**16.6891**	**19.9832**	26.2245	**18.7313**	**19.9094**	30.4392
	MAE	**21.9729**	**12.2463**	**14.5271**	18.4053	**14.2437**	**13.6255**	20.6569
	MAPE	**13.2454**	**11.5523**	13.4027	10.9286	**12.0946**	12.5402	15.2520
XGBoost-S	RMSE	34.5123	18.3918	22.1078	26.1455	21.2394	21.7922	29.5969
	MAE	24.1088	13.3061	15.4927	18.4941	15.0558	14.1222	19.6536
	MAPE	14.1671	12.1188	13.7593	10.8694	12.6695	15.9026	14.4065
XGBoost-S-lag	RMSE	32.2435	17.2803	20.9974	26.6617	20.2683	21.0335	31.4545
	MAE	22.2236	12.4192	14.5295	18.5168	14.4424	13.7168	20.8345
	MAPE	13.5993	11.5780	**13.2273**	11.1660	12.5718	13.7145	15.4466

Table 5: Down-direction-related XGBoost models' prediction results.

Methods	Error	Road 8	Road 9	Road 10	Road 11	Road 12	Road 13	Road 14
XGBoost-I	RMSE	27.6826	19.5893	18.5488	31.1616	18.1484	17.9374	**31.6893**
	MAE	19.7969	14.3576	13.3581	21.9113	13.2407	11.7431	**21.9664**
	MAPE	11.0912	12.6743	12.8101	13.7022	12.1961	**13.7475**	14.6207
XGBoost-I-lag	RMSE	**24.2552**	**17.8684**	**17.4214**	**29.0122**	**18.0647**	**17.7709**	31.9440
	MAE	**17.5001**	**12.7666**	**12.5606**	**20.4316**	**13.2205**	**11.7270**	22.2563
	MAPE	**9.7786**	**11.2418**	**11.9682**	**12.5774**	**12.1162**	14.1016	**14.5804**
XGBoost-S	RMSE	28.2182	20.2276	18.9394	32.1894	18.7528	19.9912	32.6606
	MAE	19.5314	14.3086	13.3500	22.1994	13.3074	11.8262	22.4856
	MAPE	11.0098	12.6626	12.8490	13.8508	12.3134	14.9592	14.9703
XGBoost-S-lag	RMSE	25.2074	18.7069	18.1568	30.3098	19.1776	20.2706	32.8841
	MAE	17.3920	13.0054	12.7606	20.9290	13.6335	11.9992	22.6842
	MAPE	9.8163	11.4915	12.2118	12.7907	12.4502	14.7001	14.8923

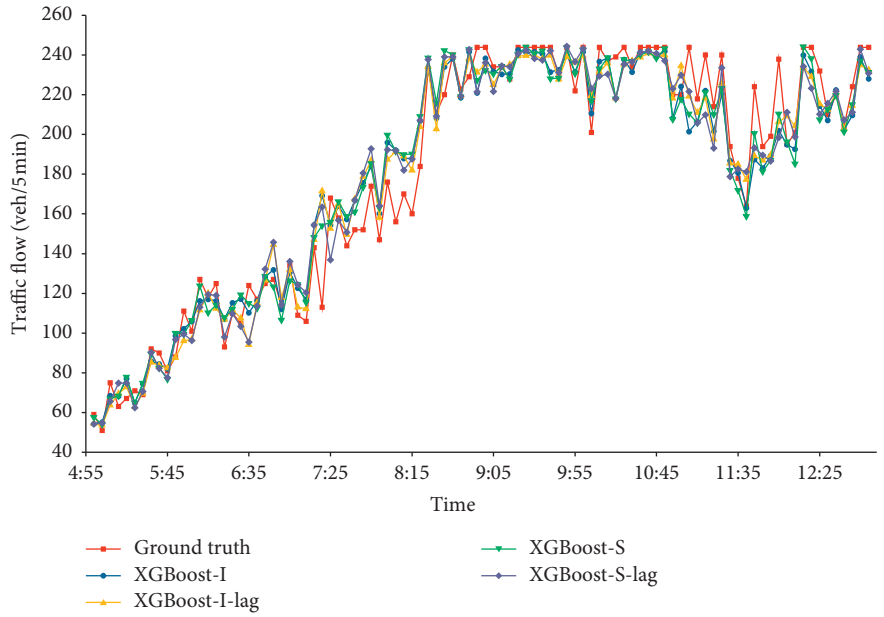
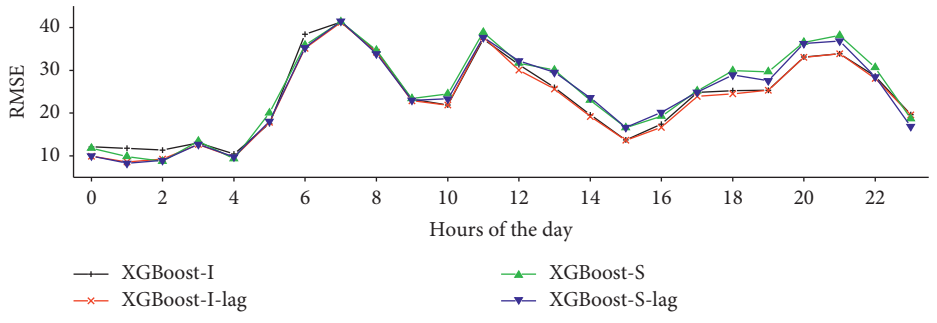
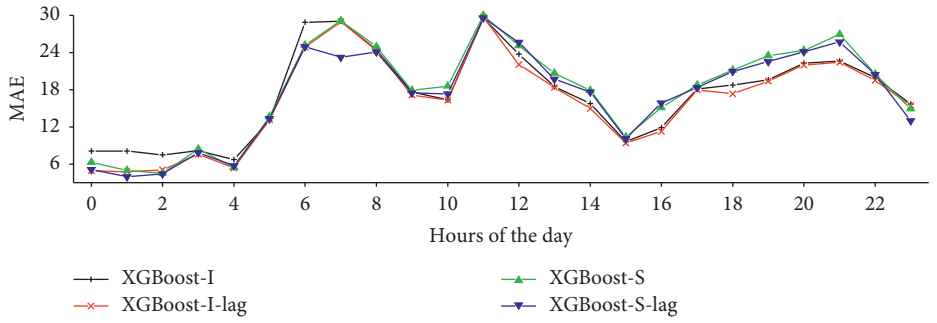


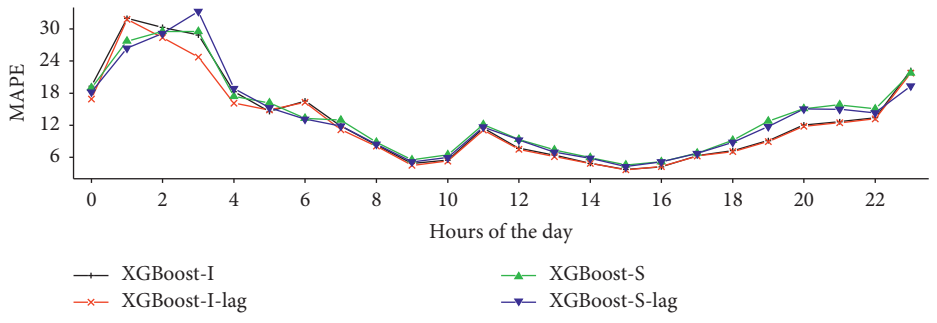
Figure 9: The four XGBoost models on the first-day predicted results and ground truth.



(a)



(b)



(c)

Figure 10: Continued.

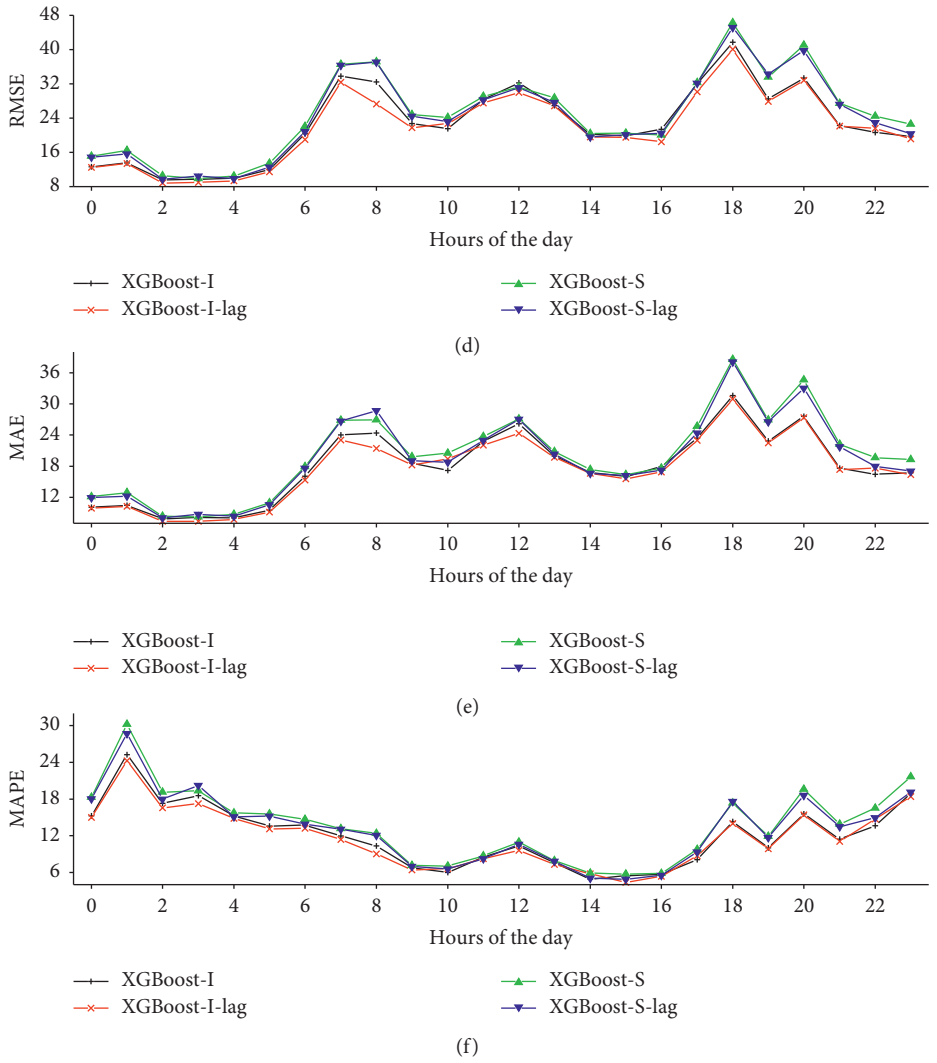


Figure 10: Three errors for the up-direction and down-direction of twenty-four hours' prediction.

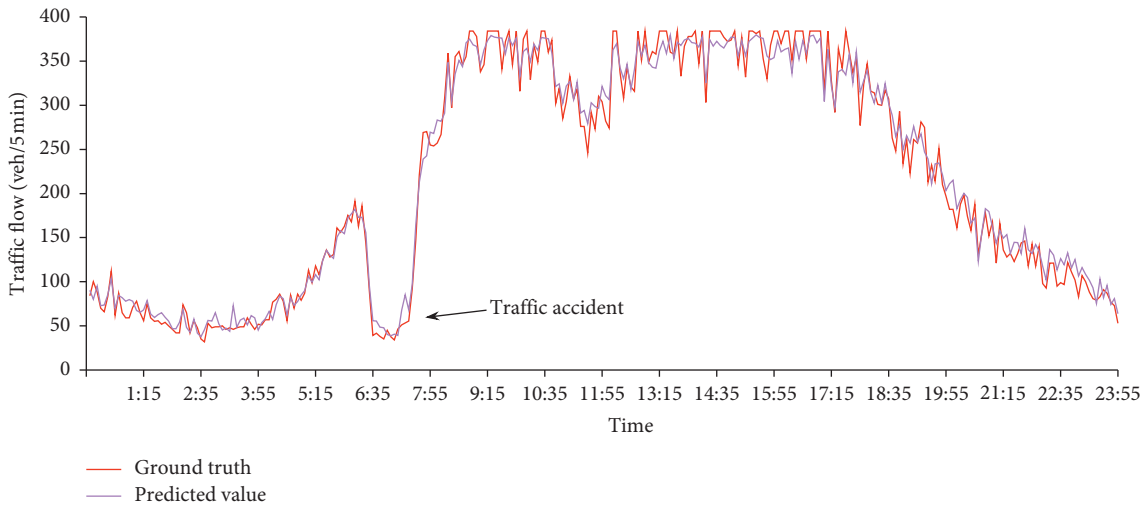


Figure 11: Special sample (traffic accident) traffic flow prediction results of the XGBoost-I-lag method.

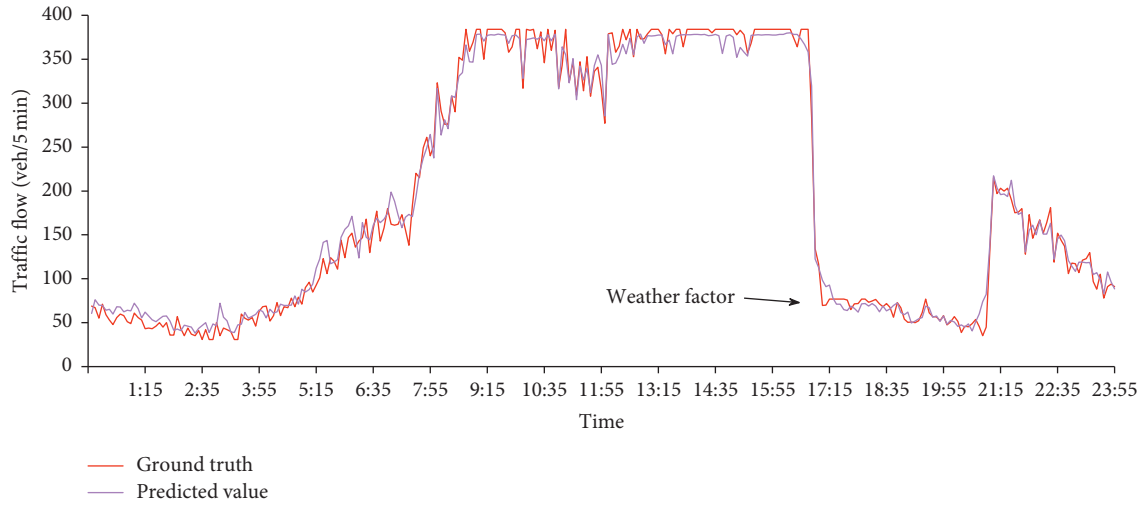


Figure 12: Special sample (weather factor) traffic flow prediction results of the XGBoost-I-lag method.

prediction in order to achieve adequate prediction results. Therefore, the XGBoost-I-lag model predicts various sudden events in dynamic traffic systems and possesses excellent prediction performance.

5.6. Missing Rate. In intelligent transportation systems, data missing is an inevitable and widespread phenomenon, although many studies [4] have been conducted in missing traffic data prediction recently, improving the real performance of recovered data sufficiently, efficiently, and accurately. We test the general problem against our model in this investigation. In case of different data missing rates, the performance of the model is further verified. Many reasons exist in the traffic flow, such as problems of the sensors, manual shutdown of the system, or signal transmission errors. Our complete datasets are divided into the following two cases of random missing data in order to detect the predictive performance of XGBoost-I-lag for missing data. One case is short-term missing, which lasts 30 minutes. The missing is mainly due to unstable equipment or chaotic environment. The other case is long-term missing, which lasts for several hours or days. Missing conditions are mainly caused by system shutdown. Average results are on RMSE and MAE error results for the cases where up-direction and down-direction datasets are missing of 10%, 20%, 30%, and 40%, which are shown in Figures 13 and 14 . We believe that over 50% of the missing data would confer difficulty in exploring the laws of transmission in this study. With the increase of the missing rate from 10% to 40%, RMSE increases from over 42% to more than 72%. For MAE, it increases from over 45% to more than 75%.

Judging from the average results of up-direction and down-direction, as the missing rate gradually increases, the error suddenly increases. When the missing rate increases to 40%, RMSE and MAE also increase by more than 70%. It is obvious that missing data has a great impact on the results of XGBoost-I-lag traffic flow prediction. Though it performs well in full data, the importance of data preprocessing is seen to have a substantial influence on the accuracy of the model as well as the prediction results.

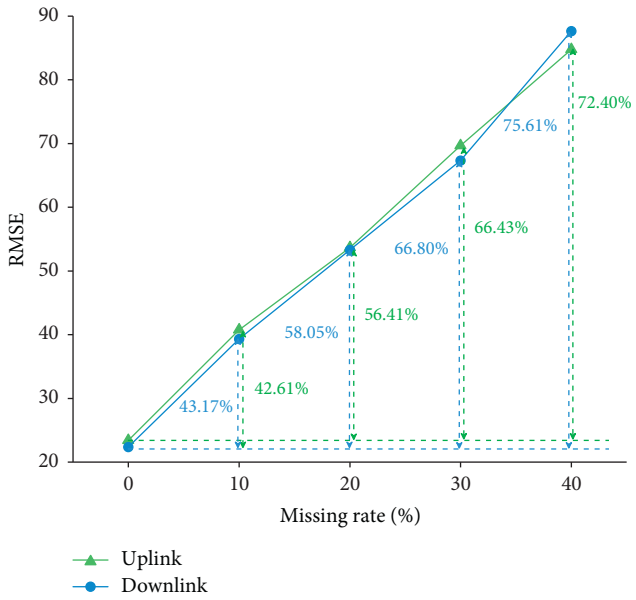


Figure 13: RMSE of up-direction and down-direction with different missing rates.

5.7. Baseline Methods. The proposed method in this study is compared with the following two baselines:

SARIMA: seasonal autoregressive integrated moving average is particularly applied in time series analysis, such as traffic flow and stock, whose data show evidence of nonstationarity, where an initial differencing step can be applied one or more times to eliminate nonstationarity. The datasets are summarized at 5-minute intervals, and the SARIMA (1, 0, 0) (0, 0, 1, 12) model is used to predict future 5-minute intervals of data.

CNN: convolutional neural network uses convolution layers with filters to extract local features by sliding windows, which can model nearby or larger spatial dependencies. It has been effectively used in traffic flow prediction and has achieved significant results in

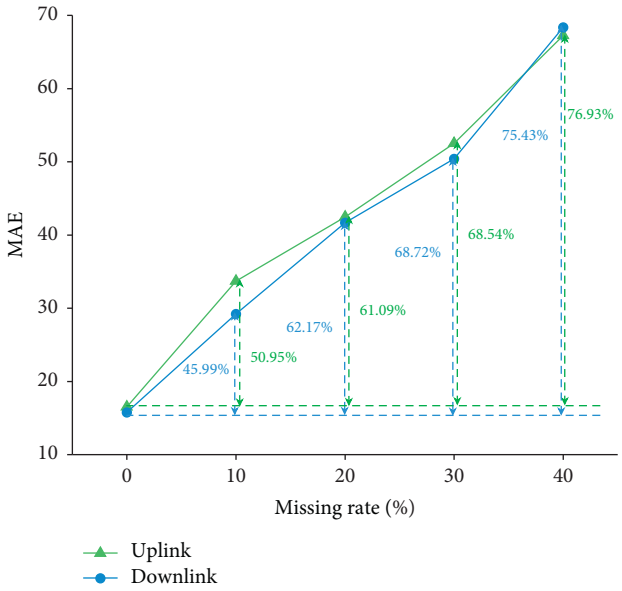


Figure 14: MAE of up-direction and down-direction with different missing rates.

capturing the long-term time dependence of the traffic flow. The network structure of CNN used in this study is described in Table 6.

RF: random forest is an algorithm fused by many decision trees, belonging to the algorithm of the bagging framework. Using the integrated idea, random forest combines multiple decision trees to improve the accuracy of classification, where each decision tree is a basic classifier. The training of each decision tree model can be extracted using self-help sampling, which randomly selects a subset from all features to train the model. Based on the classification results of all decision trees, the prediction results of the model are obtained through voting. The learning rate of the random forest is set to 0.25, the number of trees is set to 80, and other parameters are set according to their default values.

LSTM: long short-term memory network is excellent variant models of RNN, inheriting most characteristics from RNN models. It uses a finite sequence to predict traffic based on historical traffic data, which is a typical deep learning method for time series prediction. LSTM is suitable in dealing with problems that are highly related to time series. It can fit sequence data and solve problems pertaining to vanishing gradients by forgetting gate and output gate. The network structure of each LSTM is shown in Table 7.

5.8. Comprehensive Results. We compare the performance of XGBoost-I and XGBoost-S with the four baseline methods (SARIMA, CNN, RF, and LSTM) based on the datasets. Figure 15 depicts the error results of different methods as well as their corresponding spatial lag.

Compared to traditional prediction methods, XGBoost is found to perform better. Except SARIMA, the spatial lag input of the other methods is better than the ordinary input.

Table 6: CNN network structure.

Layer (type)	Output shape	Param #
input_1 (InputLayer)	(None, 6, 7)	0
conv1d_1 (Conv1D)	(None, 6, 7)	56
conv1d_2 (Conv1D)	(None, 6, 7)	56
conv1d_3 (Conv1D)	(None, 6, 7)	56
conv1d_4 (Conv1D)	(None, 6, 7)	56
conv1d_5 (Conv1D)	(None, 6, 7)	56
conv1d_6 (Conv1D)	(None, 6, 7)	56
flatten_1 (flatten)	(None, 42)	0
dense_1 (dense)	(None, 12)	516

Table 7: Each LSTM network structure.

Layer (type)	Output shape	Param #
lstm_1 (LSTM)	(None, 6, 7)	252
dense_1 (dense)	(None, 64)	3200
dense_2 (dense)	(None, 12)	780

However, SARIMA explores the individual prediction in each road without reflecting the characteristics in lag input. Instead, due to misalignment of the data, the prediction effect is evidently reduced. In addition, calculation time is based on the training and testing time. Running time of programs determines CPU time of the system. The longer the run time is, the more resources the CPU uses. XGBoost-I provides the best performance with the runtime of 162 s, though XGBoost-S gets the least running time of only 123 s. This is due to the number of trees for different roads of XGBoost-I that is different, which maximizes optimization. Therefore, an additional number of branches are explored, with the time being longer than XGBoost-S. The RF completion time is 214 s, which also serves as an ideal method in view of the results. Supporting parallel training of random forests can speed up training and is also suitable for high-dimensional data processing. Although the running time of CNN (225 s) is close to RF, its prediction is much worse. The traditional prediction methods SARIMA (383 s) and LSTM (2827 s) have longer running time. Although LSTM acquires satisfactory results, time costs and system consumption are too much. Therefore, XGBoost-I is considered to be the best choice among these six common methods for highway traffic prediction.

We utilize historical data for the next 60 minutes to predict highway traffic in the next 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, and 60 minutes. Figure 16 presents the corresponding average results of the 12 methods in short-term 5-minute steps (5, 10, 15, 20, 25, and 30).

We compare XGBoost family models with baselines models (SARIMA, CNN, RF, and LSTM) and the corresponding different input modes. Accordingly, short-term traffic flow prediction results of RMSE, MAE, and MAPE by XGBoost-I-lag are the most accurate. The reason is that XGBoost adopts different parameter adjustments and tree structures for different sections when considering temporal and spatial characteristics. Moreover, it can be observed that, in short-term traffic flow prediction, the spatial lag input of different methods is better than the results of ordinary input.

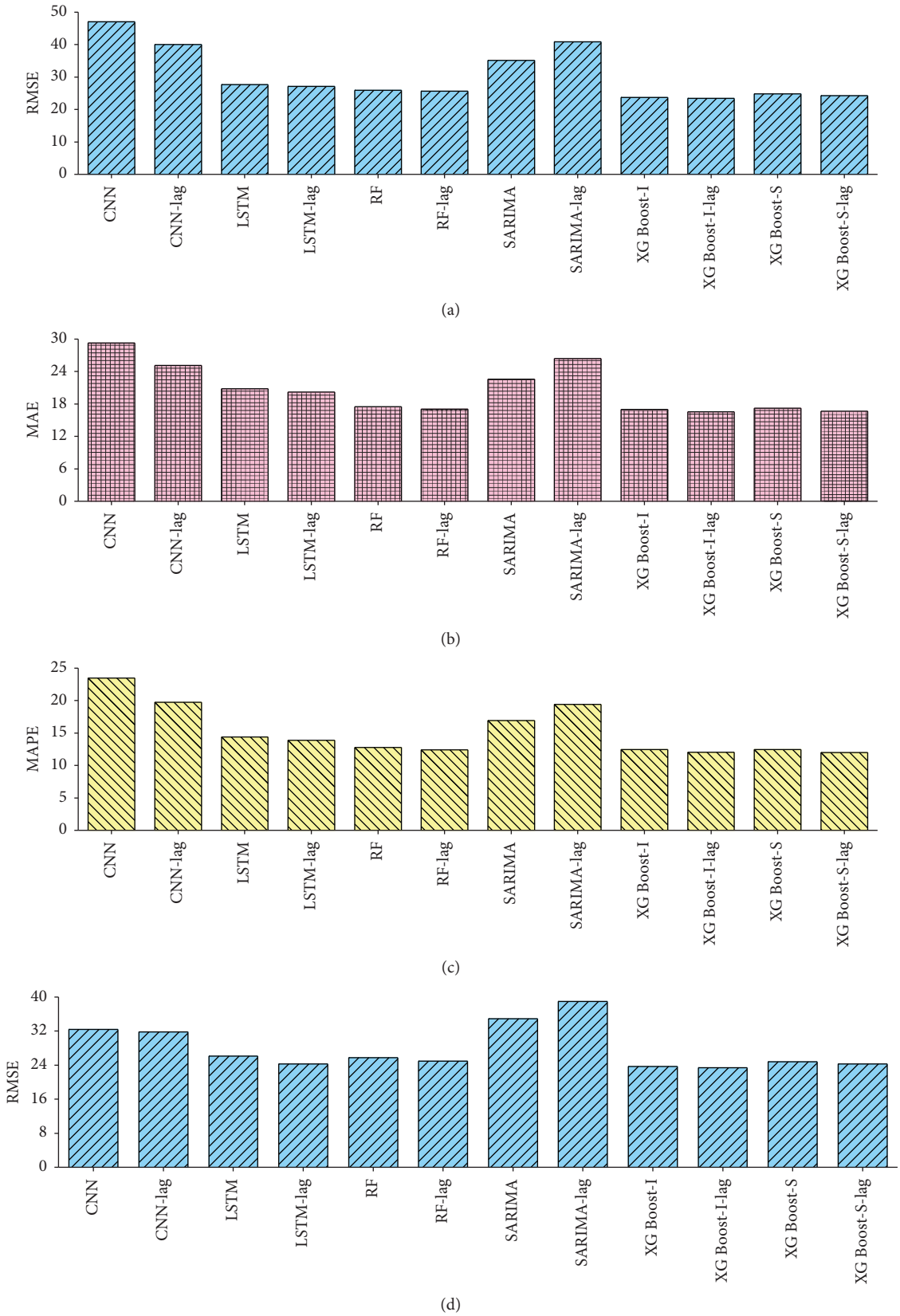
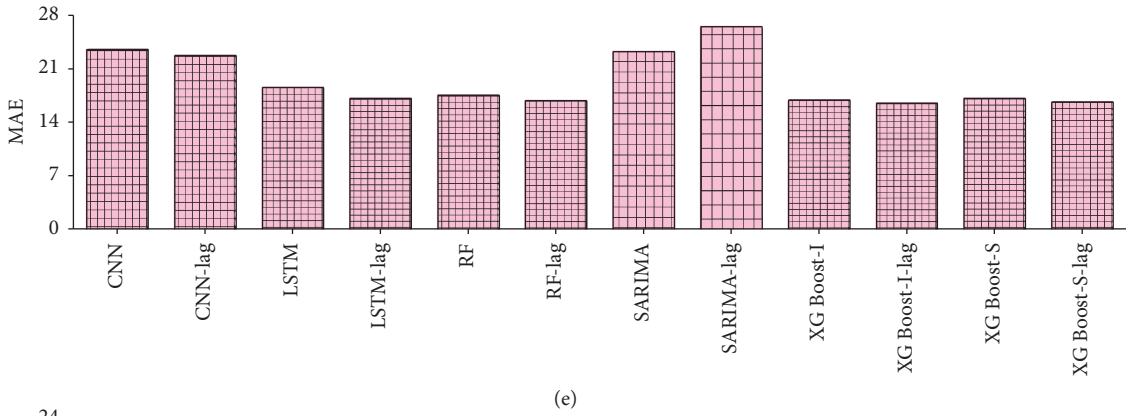
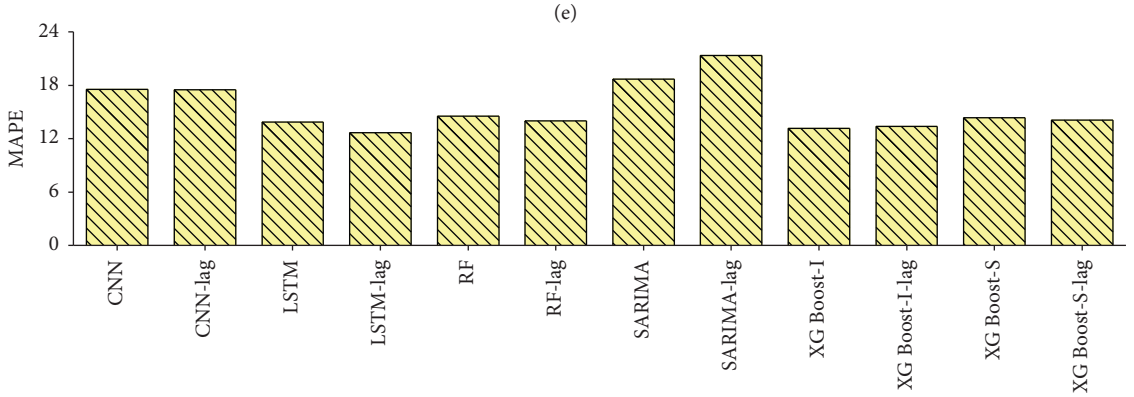


Figure 15: Continued.

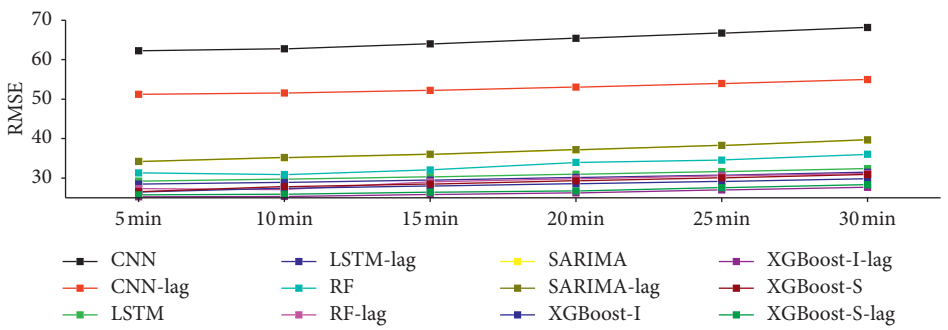


(e)

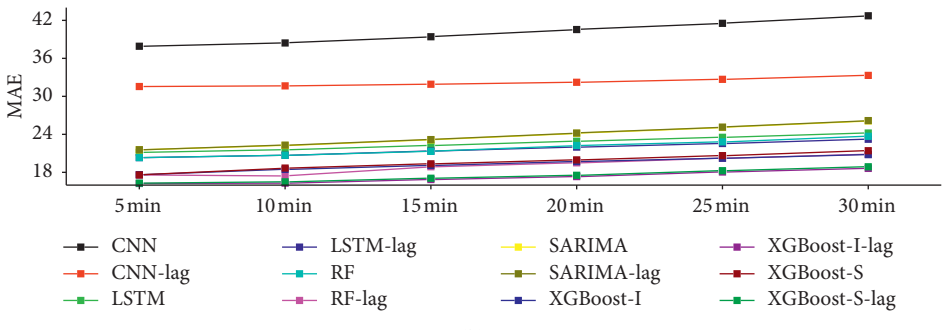


(f)

Figure 15: Comparison of three errors in different methods of up-direction and down-direction.



(a)



(b)

Figure 16: Continued.

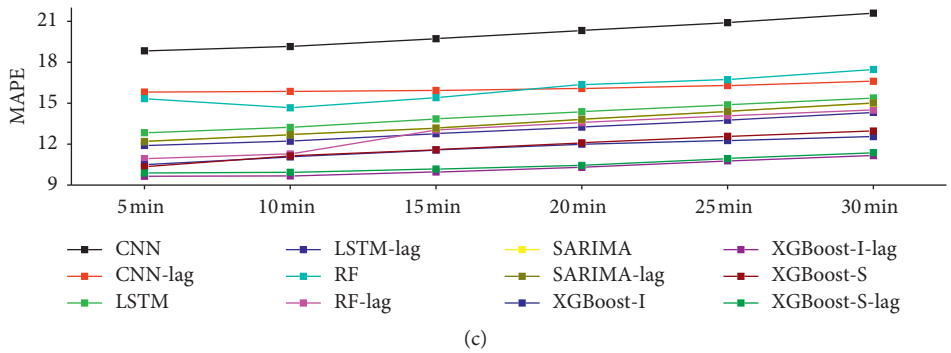


Figure 16: Comparison of traffic prediction performance of different methods in short-term 5-minute time steps of mean RMSE, MAE, and MAPE.

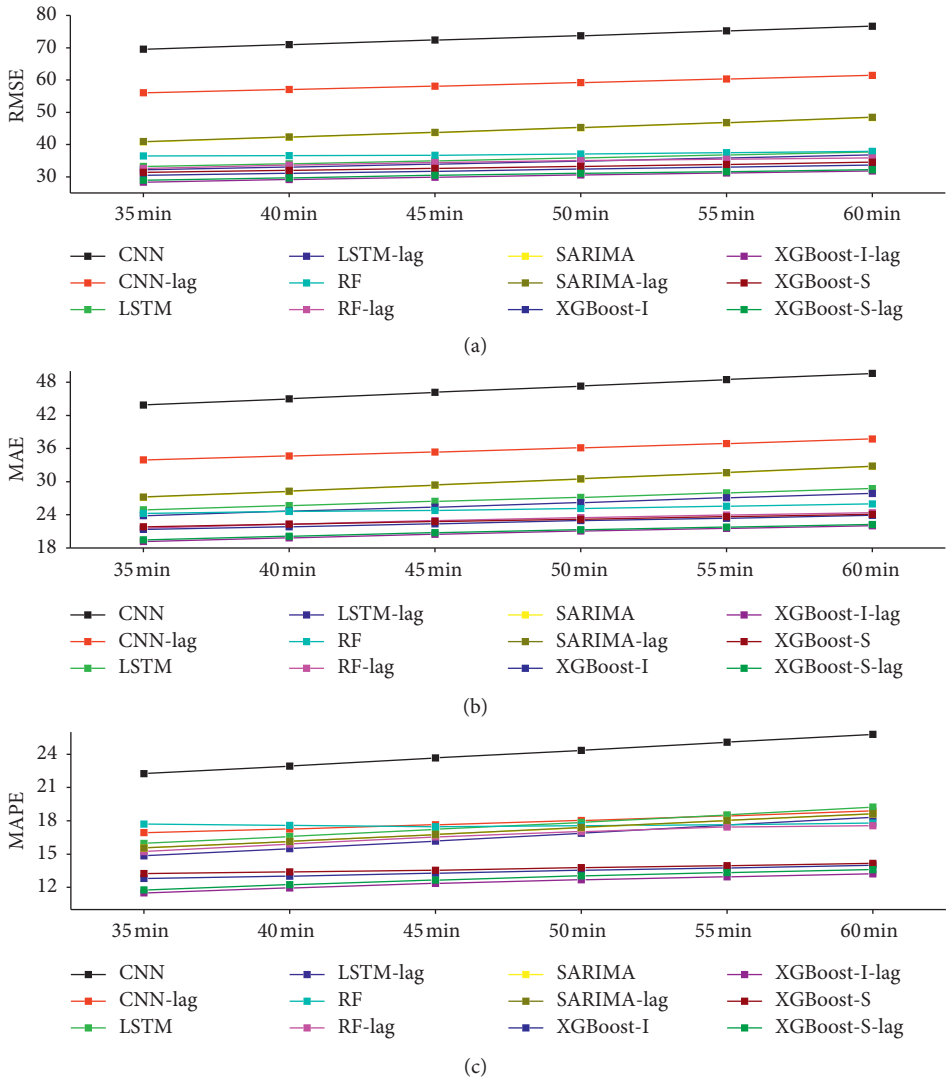


Figure 17: Comparison of traffic prediction performance of different methods in long-term 5-minute time steps of mean RMSE, MAE, and MAPE.

CNN shows the worst prediction ability, while RF and LSTM show similar accuracy, indicating that spatio-temporal characteristics play vital roles in short-term traffic prediction. The prediction errors in each method increase synchronously as the prediction range increases. Different XGBoost methods have more stable prediction trends than other methods.

Long-term forecasting is mainly contributed to travelers who plan for longer trips that are considered to be more

challenging than short-term forecasting. We predict the traffic flow for the next (35, 40, 45, 50, 55, 60) minutes based on the historical data. Figure 17 presents the results of XGBoost-I, XGBoost-S, SARIMA, CNN, RF, LSTM, as well as the corresponding spatial lag.

Regarding CNN-lag, the spatial lag input, which can be better to highlight the ability of spatial features, is evidently far superior to CNN in terms of long-term traffic flow prediction. For long-term traffic prediction, spatial information contributes better than temporal characteristics. Additionally, the advantages of CNN in utilizing the spatial characteristics in the traffic network are confirmed. The spatial lag results among the other methods are better apparently. Similar to short-term prediction, CNN performs the worst prediction performance. At the same time, RF still performs similarly to LSTM and the errors increase as the prediction range increases. However, the long-term predictive performance is marginally faster than the short-term performance. Compared with other models, XGBoost-I-lag achieves the best accuracy in both short-term and long-term of highway traffic flow prediction and obtains the most stable trend. These results prove the superiority and feasibility of the improved XGBoost model with proposed EAM optimization mode and tree structures, and the model is able to capture the traffic features and the regularity of the highway traffic flow.

6. Conclusion and Future Research

The ability of predicting the highway traffic flow in an accurate manner is important in proactive traffic management strategies in order that it can provide reliable travel information for commuters. In this paper, improved XGBoost traffic flow prediction methods and a generalized segmented-data acquirement mode are proposed. Then, we introduce an optimization way based on the EAM mode and a lag strategy involving spatio-temporal delivery. For the computing and processing datasets, the XGBoost-I parameter structures are adjusted corresponding to up-direction and down-direction roads separately. XGBoost-I-lag achieves the best performance compared with XGBoost-S series models and other baseline models. Multistep performance is evaluated, and the model is examined under the predicting of segment data and ANPRS data to prove the accuracy. It is confirmed that the missing data greatly affects the traffic flow prediction results in the XGBoost-I-lag. Except for SARIMA, the spatial lag input of all methods is better than the ordinary input. It is also observed that the identified spatio-temporal lag strategy is extremely necessary of highway traffic prediction.

In the near future, we plan to improve the predicted accuracy of the improved XGBoost framework in the following two directions: (1) more effective XGBoost parameters are worth exploring and adjusting and further expanding the usability of the EAM optimization mode. (2) Extensive segmented data calculation mode should explore more practical scenarios to divide sections subtly, and we also plan to broaden this research to estimate wider highway.

Data Availability

The ANPRS data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

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

Acknowledgments

This research was funded by National Natural Science Foundation of China (51578040), Beijing Natural Science Foundation (8162013), and Importation and Development of High-Caliber Talents Project of Beijing Municipal Institutions (CIT&TCD20180324).

References

[1] J. W. C. Van Lint and C. Van Hinsbergen, "Short-term traffic and travel time prediction models," *Artificial Intelligence Applications to Critical Transportation Issues*, vol. 22, no. 1, pp. 22–41, 2012.

[2] B. Ran, P. J. Jin, D. Boyce, T. Z. Qiu, and Y. Cheng, "Perspectives on future transportation research: impact of intelligent transportation system technologies on next-generation transportation modeling," *Journal of Intelligent Transportation Systems*, vol. 16, no. 4, pp. 226–242, 2012.

[3] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014.

[4] H. Zhang, P. Chen, J. Zheng et al., "Missing data detection and imputation for urban ANPR system using an iterative tensor decomposition approach," *Transportation Research Part C: Emerging Technologies*, vol. 107, pp. 337–355, 2019.

[5] Z. Long and Z. Zhang, "Optimization and deployment of vehicle trajectory prediction scheme based on real-time ANPR traffic big data," in *International Conference on Artificial Intelligence and Security*, pp. 74–85, Springer, Cham, Switzerland, May 2020.

[6] J. Liu, F. Zheng, H. J. van Zuylen, and J. Li, "A dynamic OD prediction approach for urban networks based on automatic number plate recognition data," *Transportation Research Procedia*, vol. 47, pp. 601–608, 2020.

[7] Y. Li, "Short-term prediction of motorway travel time using ANPR and loop data," *Journal of Forecasting*, vol. 27, no. 6, pp. 507–517, 2008.

[8] A. H. F. Chow, A. Santacreu, I. Tsapakis, G. Tanasaranond, and T. Cheng, "Empirical assessment of urban traffic congestion," *Journal of Advanced Transportation*, vol. 48, no. 8, pp. 1000–1016, 2014.

[9] K.-C. Chu, R. Saigal, K. Saitou et al., "Real-time traffic prediction and probing strategy for Lagrangian traffic data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 497–506, 2019.

[10] P. Duan, G. Mao, W. Liang, and D. Zhang, "A unified spatio-temporal model for short-term traffic flow prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 9, pp. 3212–3223, 2019.

[11] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[12] J. Chen, K. H. Low, Y. Yao, and P. Jaillet, "Gaussian process decentralized data fusion and active sensing for spatiotemporal traffic modeling and prediction in mobility-on-demand systems," *IEEE Transactions on Automation Science and Engineering*, vol. 12, no. 3, pp. 901–921, 2015.

[13] D. J. Sun, X. Liu, A. Ni, and C. Peng, "Traffic congestion evaluation method for urban arterials," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2461, no. 1, pp. 9–15, 2014.

[14] B. Sun, T. Sun, Y. Zhang, and P. Jiao, "Urban traffic flow online prediction based on multi-component attention mechanism," *IET Intelligent Transport Systems*, vol. 14, no. 10, pp. 1249–1258, 2020.

[15] M. Van Der Voort, M. Dougherty, and S. Watson, "Combining kohonen maps with arima time series models to forecast traffic flow," *Transportation Research Part C: Emerging Technologies*, vol. 4, no. 5, pp. 307–318, 1996.

[16] M. Levin and Y. D. Tsao, "On forecasting freeway occupancies and volumes (abridgment)," *Transportation Research Record*, vol. 773, 1980.

[17] H. Sun, H. X. Liu, H. Xiao, R. R. He, and B. Ran, "Use of local linear regression model for short-term traffic forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1836, no. 1, pp. 143–150, 2003.

[18] P. Jiao, R. Li, and T. Sun, "Three revised kalman filtering models for short-term rail transit passenger flow prediction," *Mathematical Problems in Engineering: Theory, Methods and Applications*, vol. 2016, Article ID 9717582, 10 pages, 2016.

[19] Y. Zhou and Z. Zhang, "Prediction of traffic flow based on deep learning," *International Journal of Advanced Computer Technology*, vol. 9, no. 2, pp. 5–11, 2020.

[20] J. Kuang, D. Zhai, X. Wu et al., "A network traffic prediction method using two-dimensional correlation and single exponential smoothing," in *Proceedings of International Conference on Communication Technology*, pp. 403–406, Innsbruck, Austria, January2013.

[21] B. M. Williams, "Multivariate vehicular traffic flow prediction: evaluation of ARIMAX modeling," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1776, no. 1, pp. 194–200, 2001.

[22] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[23] B. Ghosh, B. Basu, and M. Mahony, "Multivariate short-term traffic flow forecasting using time-series analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 10, no. 2, pp. 246–254, 2009.

[24] Y. Kamarianakis and P. Prastacos, "Forecasting traffic flow conditions in an urban network: comparison of multivariate and univariate approaches," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1857, no. 1, pp. 74–84, 2003.

[25] H. Hongqiong and T. Tianhao, "Short-term traffic flow forecasting based on ARIMA-ANN," in *Proceedings of the International Conference on Control and Automation*, pp. 2370–2373, Tokyo Japan, October 2007.

[26] J. Wang and Q. Shi, "Short-term traffic speed forecasting hybrid model based on chaos-wavelet analysis-support vector machine theory," *Transportation Research Part C: Emerging Technologies*, vol. 27, pp. 219–232, 2013.

[27] Y. Zhang and Y. Liu, "Traffic forecasting using least squares support vector machines," *Transportmetrica*, vol. 5, no. 3, pp. 193–213, 2009.

[28] L. Zheng, H. Huang, C. Zhu, and K. Zhang, "A tensor-based K-nearest neighbors method for traffic speed prediction under data missing," *Transportmetrica B: Transport Dynamics*, vol. 8, no. 1, pp. 182–199, 2020.

[29] L. Zhang, Q. Liu, W. Yang, N. Wei, and D. Dong, "An improved K-nearest neighbor model for short-term traffic flow prediction," *Procedia - Social and Behavioral Sciences*, vol. 96, pp. 653–662, 2013.

[30] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: bayesian combined neural network approach," *Journal of Transportation Engineering*, vol. 132, no. 2, pp. 114–121, 2006.

[31] Y. Lv, Y. Duan, W. Kang et al., "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[32] L. Li, L. Qin, X. Qu, J. Zhang, Y. Wang, and B. Ran, "Day-ahead traffic flow forecasting based on a deep belief network optimized by the multi-objective particle swarm algorithm," *Knowledge-Based Systems*, vol. 172, pp. 1–14, 2019.

[33] J. Tang, F. Liu, Y. Zou, W. Zhang, and Y. Wang, "An improved fuzzy neural network for traffic speed prediction considering periodic characteristic," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2340–2350, 2017.

[34] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[35] J. Gao, Y. L. Murphey, J. Yi, and H. Zhu, "A data-driven lane-changing behavior detection system based on sequence learning," *Transportmetrica B: Transport Dynamics*, vol. 1, pp. 1–18, 2020.

[36] J. Wang, R. Chen, and Z. He, "Traffic speed prediction for urban transportation network: a path based deep learning approach," *Transportation Research Part C: Emerging Technologies*, vol. 100, pp. 372–385, 2019.

[37] L. Li, B. Ran, J. Zhu, and B. Du, "Coupled application of deep learning model and quantile regression for travel time and its interval estimation using data in different dimensions," *Applied Soft Computing*, vol. 93, Article ID 106387, 2020.

[38] X. Ma, H. Yu, Y. Wang et al., "Large-scale transportation network congestion evolution prediction using deep learning theory," *PloS One*, vol. 10, no. 3, 2015.

[39] T. Chen and C. Guestrin, "XGBoost: a scalable tree boosting system," *Knowledge Discovery and Data Mining*, vol. 10, pp. 785–794, 2016.

[40] J. Cheng, G. Li, and X. Chen, "Research on travel time prediction model of freeway based on gradient boosting decision tree," *IEEE Access*, vol. 7, pp. 7466–7480, 2018.

[41] S. Yang, J. Wu, Y. Du et al., "Ensemble learning for short-term traffic prediction based on gradient boosting machine," *Journal of Sensors*, vol. 2017, Article ID 7074143, 15 pages, 2017.

[42] J. H. Friedman, "Greedy function approximation: a gradient boosting machine," *Annals of Statistics*, vol. 60, pp. 1189–1232, 2001.

[43] C. Adam-Bourdarios, G. Cowan, C. Germain-Renaud et al., "The Higgs machine learning challenge," *Journal of Physics: Conference Series*, vol. 664, no. 7, Article ID 072015, 2015.

[44] M. Chen, Q. Liu, S. Chen, Y. Liu, C.-H. Zhang, and R. Liu, "XGBoost-based algorithm interpretation and application on post-fault transient stability status prediction of power system," *IEEE Access*, vol. 7, pp. 13149–13158, 2019.

[45] G. Boeing, "OSMnx: new methods for acquiring, constructing, analyzing, and visualizing complex street networks," *Computers, Environment and Urban Systems*, vol. 65, pp. 126–139, 2017.

[46] F. G. Habtemichael and M. Cetin, "Short-term traffic flow rate forecasting based on identifying similar traffic patterns," *Transportation Research Part C: Emerging Technologies*, vol. 66, pp. 61–78, 2016.