Hindawi
Mathematical Problems in Engineering
Volume 2020, Article ID 6897965, 13 pages
https://doi.org/10.1155/2020/6897965

Research Article

Feature Selection and Model Fusion Approach for Predicting Urban Macro Travel Time

D. D. Li,[1] D. X. Yu,[1,2] Z. J. Qu,[2,3] and S. H. Yu[4]

[1]*Jilin University of Architecture and Technology, Changchun, China*
[2]*Jilin Engineering Research Center for Intelligent Transportation, Changchun, China*
[3]*School of Transportation, Jilin University, Changchun, China*
[4]*University of Minnesota Twin Cities, Minneapolis, Minnesota, USA*

Correspondence should be addressed to Z. J. Qu; quzj18@mails.jlu.edu.cn

Received 4 February 2020; Accepted 25 April 2020; Published 8 June 2020

Academic Editor: Elio Masciari

With the rapid growth of car ownership, traffic congestion has become one of the most serious social problems. For us, accurate real-time travel time predictions are especially important for easing traffic congestion, enabling traffic control and management, and traffic guidance. In this paper, we propose a method to predict urban road travel time by combining XGBoost and LightGBM machine learning models. In order to obtain a relatively complete data set, we mine the GPS data of Beijing and combine them with the weather feature to consider the obtained 14 features as candidate features. By processing and analyzing the data set, we discussed in detail the correlation between each feature and the travel time and the importance of each feature in the model prediction results. Finally, the 10 important features screened by the LightGBM and XGBoost models were used as key features. We use the full feature set and the key feature set as input to the model to explore the effect of different feature combinations on the prediction accuracy of the model and then compare the prediction results of the proposed fusion model with a single model. The results show that the proposed fusion model has great advantages to urban travel time prediction.

1. Introduction

Travel time is an important index of transportation management and the premise of traffic induction [1]. For city authorities, accurate travel time information can help develop better city management strategies and tools. For individual travelers, accurate travel time information can help them make better travel plans [2, 3], obtain the estimated arrival time, and replan the route or choose the best route according to personal preference so as to reach the destination faster. In terms of optimizing the road network, traffic conditions can be effectively improved and congestion can be effectively alleviated [4]. At the same time, it can also help public transportation (such as buses and trams) and freight service providers to better service planning and scheduling choices to improve service quality [5].

Therefore, accurate travel time prediction has important practical significance and application value for alleviating urban traffic congestion, avoiding resource waste, and improving road network performance.

With the popularization of GPS information collection technology and the improvement of data quality, it becomes possible to predict travel time by detecting data. By using the collected GPS positioning data, we obtain an extended and more comprehensive feature data set through in-depth mining. In combination with other feature data sets (such as weather and temperature), based on the analysis of travel time data features, we use the LightGBM model and XGBoost model to identify a series of key features and focus our research on features with high impact weight of travel time. Because a single model has different sensitivities to various data sets, we propose a method based on the fusion model to deal with the difference between the selected data features and the data set. By assigning different weights to different models, we can comprehensively study the data and fuse the advantages of each model.

In this paper, we are going to elaborate on the research work and achievements as follows:

(1) We propose a multimodel fusion method that linearly weights two prediction models: XGBoost and LightGBM, to improve prediction performance. Through comparative verification, the fusion-based approach outperforms the other single-model-based solution.

(2) In view of the limited number of data feature samples in the original data set, we conduct in-depth mining of GPS and other data, so as to obtain extended multidimensional spatiotemporal features (such as turning times, intersection numbers, and driving trajectories) and to study various factors affecting travel time.

(3) Our prediction of travel time is not for a certain section of the road, but for the entire urban road network. The aim is to extract key features from a combination of multiple data features and predict the travel time required for driving.

2. Related Work

Travel time is one of the key indicators reflecting the state of road traffic and plays a vital role in traffic management and operation decisions. Therefore, a wide range of algorithms for travel time prediction had been presented in the literature.

Existing travel time prediction methods can be roughly divided into two categories: one is an early statistical-based prediction method, and the other is a machine-learning-based prediction method. Inspired by statistics and travel time prediction requirements, many methods have been proposed in the literature, including the simple moving average method [6] and exponential smoothing method [7]. The latter is a linear model with a smoothing function over time, which is widely used in combination with sensor data. There are also methods based on Kalman filtering [8, 9], generalized autoregressive conditional heteroskedasticity (GARCH) model [10], and autoregressive integrated moving average (ARIMA) model [11]. Among them, the ARIMA model is widely used in highway traffic prediction research because of its good theoretical basis and prediction accuracy [12]. The statistical modeling methods in the above documents all assume that the data have a specific model structure. When the traffic flow changes regularly, it can get a good prediction effect according to its parameter changes. However, these methods cannot directly deal with data features of mixed types and need to transform and process multidimensional data features to be used as model input. Moreover, the model does not have a good processing method for data outliers and missing values, and the accuracy will be reduced for complex road networks.

Most of the recent predictions of travel time have adopted machine learning methods [13, 14] including k-nearest neighbor (KNN) algorithms [15, 16], support vector machines [17], and neural networks [18, 19] model. Compared to earlier statistical prediction methods, machine learning models do not assume any specific model structure for the data, but treat it as unknown, which can handle complex problems and large amounts of data well. And because of the nonlinear characteristics of traffic flow, machine learning is generally more efficient and accurate than traditional statistical methods in travel time prediction methods. However, most machine learning models lack a reasonable interpretation of the results, which also limits their application to travel time to some extent.

In recent years, with the emergence of data mining and improved machine learning models, tree-based integrated algorithms have been widely used in solving prediction and classification problems. Moreover, they have achieved good results in different fields, such as ecology [20], economics [21], bioinformatics [22, 23], and health care [24]; they are considered one of the most successful general-purpose algorithms in modern times. Rather than searching for the optimal single model, the tree-based integration algorithm can enhance the overall prediction capability by assembling multiple tree models [25], which can not only make efficient prediction but also identify and explain the interaction between data features. At the same time, it is not sensitive to monotone transformation of characteristic data input and is robust to missing value and outlier value. It can interact perfectly with feature data of mixed types and can fit complex nonlinear relationships [25]. These characteristics make the tree-based integration method good for the prediction of travel time.

However, there are relatively few applications of tree-based integration methods in the field of transportation. Ma et al. [26] proposed a method to predict traffic accidents by using different data variables, and the predicted result was better than BP neural network, support vector machine, and random forest. However, the nonlinear relationship between the influence characteristics and the predicted value was not studied. Li [27] proposed based on random forest and gradient-enhanced regression tree model that the fitting effect of data features and different models was systematically studied, and data features were combined to predict travel time. However, GPS was not deeply mined, and data features were not extensive enough, resulting in a lack of universality. In order to improve the prediction accuracy of journey time, Cheng et al. [28] proposed a trip time prediction model based on the gradient boosting decision tree (GBDT) and predicted the journey time with high accuracy by exploring the key influencing variables. The prediction results of the single model will be affected by the selection of feature variables, but the article lacks instructions on the selection of feature variables.

The above data-driven model is good at capturing the spatiotemporal relationship between data, widely used in predicting travel time methods, and has high accuracy. Similarly, with the development of artificial intelligence, the self-learning ability of deep learning can fully utilize the advantages of massive data and deeply explore the potential characteristics of traffic data. Among them, He et al. [29] use low-frequency detection vehicle data to identify the congestion of turning intersections in the road network and provide new ideas for the deep mining of road network traffic status data. Lin et al. [30] proposed a new graph convolutional neural network to predict the site-level hourly

demand in a large-scale bicycle sharing network. Liu [31] proposes a heuristic feature selection method based on cohesion for short-term traffic flow prediction. The proposed feature selection method can reduce the prediction error in different scenarios, but the method only selects one variable in the objective function. Some other factors can be considered. Zhao et al. [32, 33] established an LSTM neural network model based on spatiotemporal characteristics, respectively, predicted traffic flow and travel speed, and obtained good prediction accuracy. The selection of features shown in the article is very important for the fitting results. If more useless features are selected, it will only increase the data redundancy and complexity, and the prediction effect may not increase but decrease.

In order to better solve the above problems, we propose a method based on multimodel fusion, using different feature combinations, to study the travel time prediction performance through two popular decision tree integration methods: LightGBM and XGBoost. On the one hand, these two models are excellent in various prediction aspects and fields; on the other hand, LightGBM and XGBoost are both decision tree models with integrated frameworks.

3. The Study

3.1. XGBoost Model. XGBoost is an optimization algorithm for the boosting algorithm. It can turn a variety of weak feature classifiers into a strong classifier through an integrated algorithm. Its algorithm continuously improves the accuracy of the model by continuously superimposing and returning the residuals to resuperimpose. Therefore, the XGBoost algorithm can better predict the city travel time after processing a variety of features, reduce prediction errors, and obtain higher prediction accuracy.

For a given sample training set with N samples and M characteristics $K = \{(x_i, y_i)\}(i = 1, 2, \ldots, N, x_i \in R^M,\ y_i \in R)$, the XGBoost algorithm training process is an integrated model formed by adding K CART (classification and regression tree) functions, which is shown as follows:

$$\widehat{y}_i = \sum_{t=1}^{n} f_t(x_i), \quad f_t \in F, \tag{1}$$

where n is the number of trees, the set F is the set of all possible CARTs, f_t is a function of the set F, $\widehat{y}_i$ is the predicted value, and x_i is the i-th input data. For each given sample, the XGBoost model obtains the final prediction value $\widehat{y}_i$ by retaining $(t-1)$ rounds of model prediction in each iteration and adding a new function $f_t(x_i)$ at the end. Its iterative process is as follows:

$$\begin{aligned} \widehat{y}_i^{(0)} &= 0, \\ \widehat{y}_i^{(1)} &= f_1(x_i) = \widehat{y}_i^{(0)} + f_1(x_i), \\ &\vdots \\ \widehat{y}_i^{(t)} &= \sum_{k=1}^{t} f_k(x_i) = \widehat{y}_i^{(t-1)} + f_t(x_i). \end{aligned} \tag{2}$$

In order to measure the gap between the predicted value and the real value in machine learning, a loss function is defined to describe it, and the model is repeatedly trained to optimize the overall objective function. The objective function of the XGBoost model is as follows:

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} l(y, \widehat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{3}$$

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2. \tag{4}$$

There are many kinds of objective functions. According to different types of problem solving, the self-loss function $l(y, \widehat{y}_i)$ can choose mean square error function, logarithmic loss function, etc.

$\Omega(f_k)$ is the regularized penalty term, whose function is to prevent the overfitting phenomenon caused by too many leaf nodes. T is the number of leaves, w_j is the weight of leaf nodes, γ is the punishment intensity, and λ is the parameter to ensure that the number of leaf nodes is not too large. Generally, $\lambda = 1$, and only γ is adjusted.

According to the defined objective function and loss function, the training samples can be used to train the XGBoost model. As mentioned above, equations (3) and (4) are the end of the model iterative update process, and then the objective function can be updated as

$$J(f_t) = \sum_{i=1}^{n} \left(y_i, \widehat{y}_i^{(t-1)} + f_t(x_i)\right) + \Omega(f_t) + C. \tag{5}$$

In order to find the most suitable tree structure f_t to optimize the objective function and make it reach the optimal value, the loss function can be used to perform approximate processing on the Taylor second-order expansion at $f_t = 0$, so the objective function can be approximately expressed as

$$\mathrm{Obj}^{(t)} = \sum_{i=1}^{n} \left[L\left(y_i \widehat{y}_i^{(t-1)}\right) + g_i f_t(x_i) + \frac{1}{2} h_i f_i^2(x_i) \right] + \Omega(f_t) + C, \tag{6}$$

where $g_i = \partial_{\widehat{y}_i^{(t-1)}} l(y_i \widehat{y}_i^{(t-1)})$ and $h_i = \partial^2_{\widehat{y}_i^{(t-1)}} l(y_i \widehat{y}_i^{(t-1)})$ are the first and second derivatives of the self-loss function at the expansion point $\widehat{y}_i^{(t-1)}$, respectively; before the t-th iteration, the loss function $l(y, \widehat{y}_i^{(t-1)})$ composed of the output $\widehat{y}_i^{(t-1)}$ and sample y_i of CART function is a fixed value, and the change in its constant term C has no effect on the overall objective function. Therefore, the objective function can be simplified to

$$\mathrm{Obj}'^{(t)} = \sum_{j=1}^{T} \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) {w_j}^2 \right] + \gamma T. \tag{7}$$

Map x to each corresponding leaf node, and then j can be defined as the sample set above each leaf node. w_j is the weight of the newly established tree leaf, and the partial

derivative of it can obtain the weight of the optimal leaf node as

$$w_j^* = -\frac{G_j}{\left(H_j + \lambda\right)}, \tag{8}$$

where $G_j = \sum_{i \in I_j} g_i$ and $H_j = \sum_{i \in I_j} h_i$, y on substituting w_j^* from the above equation into the objective function, the optimal objective function corresponding to the newly established tree can be obtained as

$$J'\left(f_t\right) = \frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_j + \lambda} + \gamma T. \tag{9}$$

$J'\left(f_t\right)$ obtained from the training can be used to measure the excellence of any new tree structure. The smaller the value is, the more the loss function of the model can be reduced by the tree structure established.

3.2. LightGBM Model. LightGBM (Light Gradient Boosting Machine) is a fast, distributed, high-performance gradient boosting framework based on decision tree algorithms. It is commonly used in sorting, classification, regression, and other machine learning experiments. LightGBM speeds up training and reduces memory usage by segmenting the values of successive features (attributes). Its main features are the decision tree algorithm using histogram and leaf-wise leaf growth strategy with depth limitation.

3.2.1. The Histogram Algorithm. The basic idea of histogram algorithm is to discretize the continuous floating-point eigenvalues into k integers and construct a histogram of width k. While traversing the data, the discrete value is used as the index for traversal. After traversing the data once, the histogram counts the data quantities of various eigenvalues and then searches for the optimal segmentation point based on the discrete value of the histogram. This not only reduces the traversal complexity but also increases the running speed, as shown in Figure 1.

3.2.2. Leaf-Wise Leaf Growth Strategy. Leaf-wise is a more efficient decision tree splitting strategy. Each time it splits, it finds the optimally split leaf in the cotyledon (usually the leaf with the largest amount of data) and then splits, and so on. Therefore, compared with the original decision tree splitting strategy, in the same process of finding the optimal splitting method, leaf-wise will reduce more time consumption and has a higher accuracy. The disadvantage of leaf-wise is that it may grow deeper decision trees and produce overfitting. Therefore, LightGBM adds a maximum depth limit to leaf-wise to ensure high efficiency while preventing overfitting. The splitting strategy is shown in Figure 2. Green is the split maximum return leaf, and black is the nonsplit maximum return leaf.

3.3. Model Fusion. In Figure 3, we use the idea of combining the advantages of the two models to fuse the models. Because each model has different sensitivities to different key features, different prediction results will be produced. In this paper, we assign weights to the respective prediction results of each model and make them cooperate with each other to bring greater advantages and eliminate disadvantages so that the model as a whole shows better robustness and universality.

After obtaining the XGBoost and LightGBM prediction results through the two prediction methods described above, this paper uses the linear weighted fusion method in the following equation to optimize the prediction results:

$$\text{predict} = \omega_1 \cdot \text{predict}_{lgb} + \omega_2 \cdot \text{predict}_{xgb}. \tag{10}$$

When weighting each model, because LightGBM has a more advanced leaf splitting strategy, partitioning occurs only at the node that has the most revenue, which improves the speed of decision tree construction and is more suitable for processing short-term travel time predictions, so it should be allocated higher weight value. At the same time, we explore the weights in depth and finally choose the inverse error method to determine the weights, as shown in the following equations:

$$\omega_1 = \frac{\varepsilon_2}{\varepsilon_1 + \varepsilon_2}, \tag{11}$$

$$\omega_2 = \frac{\varepsilon_1}{\varepsilon_1 + \varepsilon_2}, \tag{12}$$

where ε_1 and ε_2 are the mean square errors of LightGBM and XGBoost. From equations (11) and (12) above, it can be seen that this method can reduce the overall error of the combined model by assigning a larger weight to the model with a relatively small error, so as to achieve the overall optimization of the model. Through relevant calculations and experiments, we give the final weight distribution and fusion model form, as follows:

$$\text{predict} = 0.8 \times \text{predict}_{lgb} + 0.2 \times \text{predict}_{xgb}. \tag{13}$$

4. Results and Discussion

4.1. Data Exploration and Analysis. In this paper, the GPS data of 11.63 million vehicles for 11 days from December 10 to 20, 2018, in Beijing are used as the basic data. The original GPS data points are matched with the actual map to filter out the required GPS of 8.56 million vehicles. Data set covers daily scoring and peak hours, which can meet the needs of analysis. By matching the data points with the actual road network, we can eliminate redundant points with errors (points that are not in the actual road network or do not match the actual research location). The effect of combining with the actual road network after screening is shown in Figure 4.

The GPS data of the same ID vehicle were collected 3–5 times in the original data set. Some data fields from the initial data set are shown in following Table 1. Among them, ID is the vehicle number, and the location speed is the instantaneous speed of the vehicle at the time of reception, and the unit is km/h. Since the spot speed is accidental, we introduce

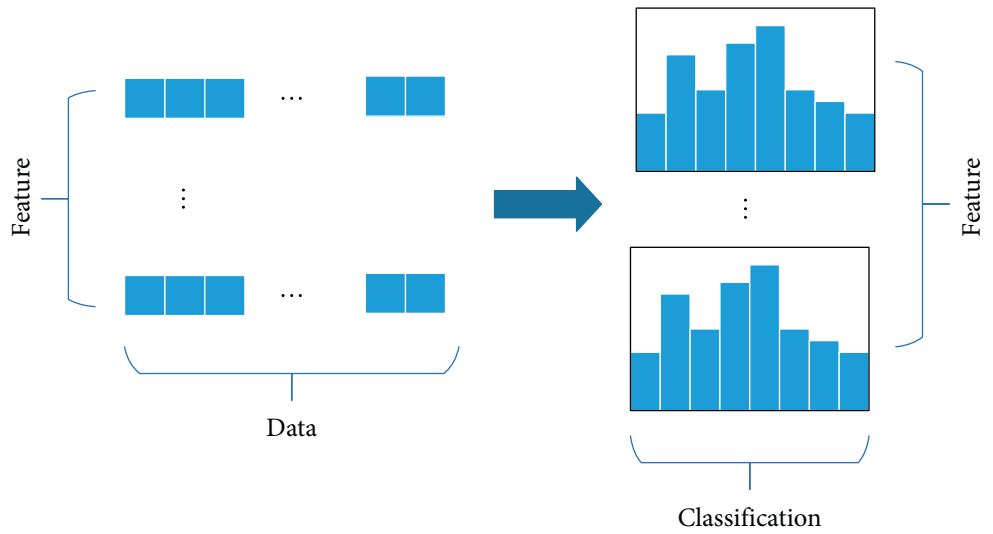

Figure 1: Schematic diagram of histogram algorithm.

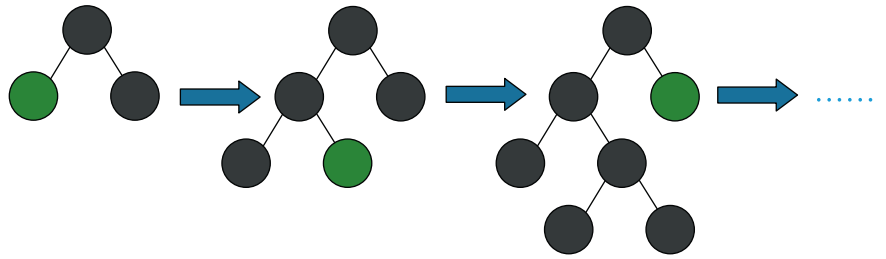

Figure 2: Leaf-wise leaf growth strategy.

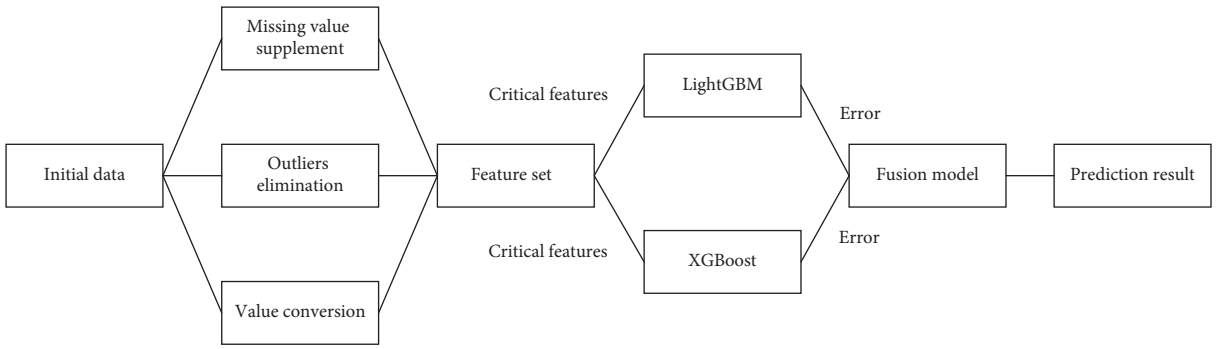

Figure 3: General solution framework.

the average value of the speed of the location measured by all GPS points of the same vehicle-mean spot speed for research.

The unit of the driving direction is degree, the due north is 0, it increases clockwise, and the value range is [0,360). The travel time required in this paper can be obtained by calculating the time difference value between the reception time of the first GPS point and the last GPS point detected by the vehicle. Vehicle type 2 is represented as a passenger car.

4.1.1. Driving Feature Mining. Through the analysis of the article above, we can see that the data features directly obtained from GPS data are relatively limited, which requires further mining and analysis. In this article, we map the multiple point data collected from the same vehicle to the actual road network to obtain data characteristics such as the number of turns, the number of intersections passed, and driving directions. As shown in Figure 5(a), D_1, D_2, and D_3 are the mappings of three consecutive GPS points in the

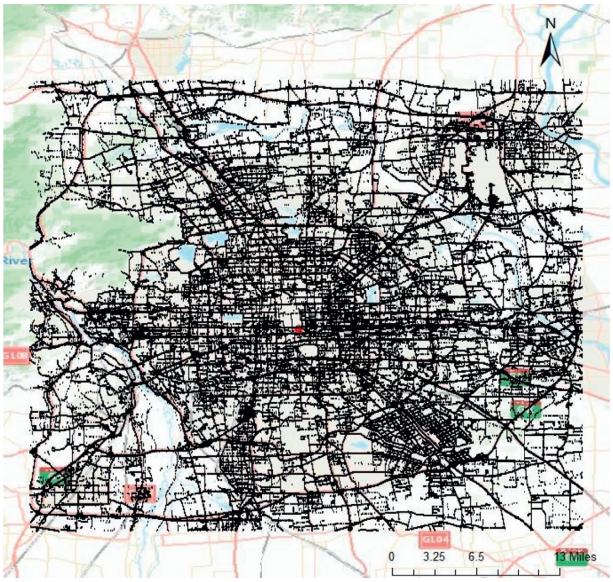

Figure 4: GPS data combined with road network map.

Table 1: GPS initial data set.

ID	Longitude	Latitude	Spot speed	Traveling direction	Receipt time	Vehicle type
42468616724	116.6104400	40.5108900	46	52	5:56:13	2
42468616724	116.6130900	40.5123400	44	52	5:57:35	2
42468616724	116.6153900	40.5136400	36	52	5:59:57	2
41675202048	116.4027200	39.1643100	12	0	5:56:02	2
41675202048	116.4027200	39.1688199	28	358	5:58:24	2
41675202048	116.4126700	39.1735600	40	358	5:59:46	2
42468116615	116.9309000	39.6290800	48	88	5:56:08	2
42468116615	116.9322100	39.6291300	24	82	5:57:29	2
42468116615	116.9325000	39.6309800	56	0	5:59:49	2
.....	.....	.....	.....	.....	.....	.....

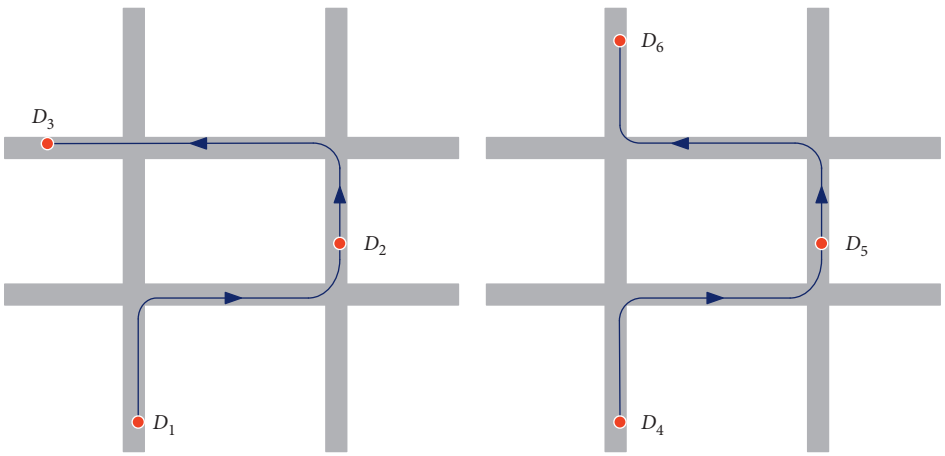

Figure 5: GPS points and roads combination diagram.

actual road network measured by the same vehicle during the detection time. According to the detection time, the driving path direction can be analyzed as $D_1 \longrightarrow D_2 \longrightarrow D_3$, the number of crossings is 4, and the number of turns is 3. Similarly, the detection direction of the vehicle in Figure 5(b) is $D_4 \longrightarrow D_5 \longrightarrow D_6$, the number of passing intersections is 4, and the number of turns is 4.

In order to match the driving conditions of the vehicle on the current road section and the degree of congestion on the road section, we combine the traffic flow in the road section where the vehicle is driving in the same time period as above. Through mining the GPS data above and combining it with the actual road network, we can also obtain the relevant characteristic variables such as the length of the driving section, space occupancy, and density. The data set after data mining is shown in Table 2.

4.1.2. Weather and Temperature Characteristics. In order to obtain a more comprehensive data set of influencing factors, we considered driving characteristics data and obtained weather, temperature, and other information of Beijing from December 10 to 20, 2018, through investigation. After this, we added it as the feature information of travel time prediction to the feature sequence, as shown in Table 3.

Table 4 divides the weather characteristics into 6 fields, where dwc represents the weather condition from 6:00 to 18:00, nwc represents the weather condition from 18:00 to 6:00 the next day, and dtv represents the temperature from 6:00 to 18:00. The value of ntv represents the value of temperature from 18:00 to 6:00 the next day; dwinds represents the wind direction and wind speed from 6:00 to 18:00, and nwinds represents the wind direction and wind speed from 18:00 to 6:00 the next day.

In this paper, we combine the driving feature data based on GPS data mining with weather and temperature feature data as the feature input data for this experiment. In order to explore the correlation between travel time and each data feature in the above data set and further explore the relationship between each data feature, we draw a corresponding scatter plot based on the data distribution (Figure 6). From the scatter plots between the various data features, we can find that there is a potential logarithmic relationship between density and average location speed, and the cumulative trends of the vehicle, density, and average location speed data are close to the parabola. Traffic flow, location speed, and average location speed are the abscissa; the curve shape of the cumulative frequency histogram can be deduced from the data volume distribution, which should also conform to the Poisson distribution, and initially confirms that the data source has a certain reliability. Since the data without processing have outliers and missing values, the scatter plot of related features alone cannot well analyze the criticality of each feature in the prediction model, so we need to analyze it further.

4.2. Data Processing and Feature Extraction

4.2.1. Data Preprocessing. From the above analysis, we can know that the original data have outliers and missing values, so the scatter plots of related features cannot well analyze the criticality of each feature to the prediction model. For these two types of abnormal data, we use the corresponding methods to deal with them. In this article, we use the idea of box plots (Figures 7(a) and 7(b)) to deal with the outliers of the data. From Figures 7(a) and 7(b), we can see that the data distribution of the travel time is more scattered. And through the box plot of density, traffic volume, and average location speed, we can find that the data distribution is more scattered. If we use the box plot method to delete the value 1.5QR higher than the upper score or 1.5QR lower than the lower quantile, some real values that can be used for prediction will be deleted by mistake, which will reduce the robustness of the model. Therefore, we set thresholds for this type of data to remove unreasonable values and set them to null values. Finally, we use data filling methods to supplement missing and null values.

Common missing value processing methods can be divided into several categories. The first type: when the number of missing data is greater than 35%, the feature column can be deleted. The second type: for a feature that lacks a single piece of data, the random forest model can be trained with the part without missing values in the feature, and then, the missing data can be filled. The third type: in certain feature data, a single value is intermittently missing, and we can linearly interpolate the corresponding feature to fill the missing value, as follows:

$$y = y_0 + \frac{y_1 - y_0}{x_1 - x_0}(x - x_0). \tag{14}$$

In equation (14), y_0, y, y_1 is a series of data points arranged in ascending order from time to time, which is the previous point closest to the missing point y in time and x_0 is the corresponding time point. Similarly, y_1 is the next point closest to y in time and x_1 is the corresponding time point.

After preprocessing the data, the obtained characteristic data table is shown in Table 4. We can see that most of the data are more in line with the actual data trend, thus completing the work on data screening and processing, and this feature data table is used as an example of a training/test file for the model.

4.2.2. Analysis of Key Features. Different models generally use different methods to judge the importance of each data feature on the prediction result. For example, LightGBM is different from the traditional method of category splitting. It uses the "many vs many" splitting method and reclassifies the categories according to the relevance of the training target. More specifically, the category features can be considered discretized. For subfeatures, LightGBM reorders each subfeature according to the weight and then selects the optimal segmentation point. XGBoost, on the other hand, uses the average gain of each feature to evaluate the importance of the feature. If the average gain is higher, the feature is more important.

Figures 8 and 9 show that the LightGBM and XGBoost models are ranked according to the importance of data

Table 2: Data set after data mining.

ID	Spot speed	Mean spot speed	Traveling direction	Vehicle type	Density	Space occupancy	...
42468616724	75	53	52	2	0.07	0.23	...
42468616724	76	37	52	2	0.09	0.62	...
42468616724	38	38	52	2	0.14	0.48	...
41675202048	51	42	0	2	0.12	0.3	...
41675202048	41	45	358	2	0.11	0.3	...
41675202048	57	28	358	2	0.17	0.46	...
42468116615	56	28	88	2	0.14	0.43	...
42468116615	77	40	82	2	0.13	0.91	...
42468116615	90	43	0	2	0.08	0.25	...
...	...	...	...	...	...	...	

Table 3: Weather feature data set.

Data	dwc	nwc	dtv	ntv	dwinds	nwinds
2018.12.10	Cloudy	Sunny	7	0	Northeast wind force 1~2	Northeast wind force 1~2
2018.12.11	Cloudy	Sunny	8	2	Northeast wind force 1~2	Northeasterly breeze
2018.12.12	Fog	Cloudy	8	−3	Northeast wind force 3~4	Northeast wind force 3
2018.12.13	Sunny	Sunny	4	−6	Northwest wind force 1~2	Northwesterly breeze
2018.12.14	Rain	Cloudy	0	−4	Southwesterly breeze	Southwesterly breeze
2018.12.15	Fog	Sunny	−2	−9	Northwest wind force 3~4	Northwest wind force 3~4
2018.12.16	Sunny	Cloudy	−5	−10	Northwest wind force 3~4	Northwesterly breeze
2018.12.17	Sunny	Sunny	−2	−10	Southwest wind force 2~3	Southwesterly breeze
2018.12.18	Cloudy	Sunny	−1	−10	Southeast wind force 1~2	Southeasterly breeze
2018.12.19	Cloudy	Sunny	0	−7	Northeast wind force 1~2	Northeast wind force 1~2
2018.12.20	Sunny	Cloudy	−1	−10	Northwest wind force 2~3	Northwesterly breeze

features. Both models point out that the most critical feature is the mean spot speed. From the overall analysis, it can be seen that the spatial features such as traffic flow, density, driving direction, and space occupancy have a stronger impact on the overall forecast. In contrast, environmental features such as weather conditions and temperature have less impact.

In addition, the recorded value of travel time in this article varies between 4 and 11 min, which is a short-term prediction. Therefore, the volatility of the data may have a certain impact on the weight of the data features. Features such as mean spot speed, traffic flow, and driving direction that have large numerical fluctuations will have a greater impact on the prediction of travel time. Conversely, features such as number of intersections, number of turns, and weather conditions have less numerical fluctuations. The prediction of travel time will have a smaller effect.

4.2.3. Comparative Analysis of Model Prediction Results. In this paper, we choose root mean square error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and symmetric mean absolute percentage error (SMAPE) as the evaluation indicators for this experiment. They all obey the same rule: when the predicted value completely matches the true value, the error is zero, and the model predicts the best effect; the larger the error, the larger the calculated value.

The definition and formula of each indicator are as follows:

Predictive value: $\hat{y} = \{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_n\}$

Actual value: $y = \{y_1, y_2, \ldots, y_n\}$

Root mean square error (RMSE):

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}. \tag{15}$$

Mean absolute error (MAE):

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|\hat{y}_i - y_i\right|. \tag{16}$$

Mean absolute percentage error (MAPE):

$$\mathrm{MAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{y}_i - y_i}{y_i}\right|. \tag{17}$$

Symmetric mean absolute percentage error (SMAPE):

$$\mathrm{SMAPE} = \frac{100\%}{n}\sum_{i=1}^{n}\frac{\left|\hat{y}_i - y_i\right|}{\left(\left|\hat{y}_i\right| + \left|y_i\right|\right)/2}. \tag{18}$$

When verifying the results, the data of December 10–18 of the data set are used as the training set for the model, and the data of December 19–21 are used as the test set for the model. In the training process, the traditional GBDT model was added as a comparison, and the early termination strategy was adopted to prevent the occurrence of overfitting. In order to further verify whether

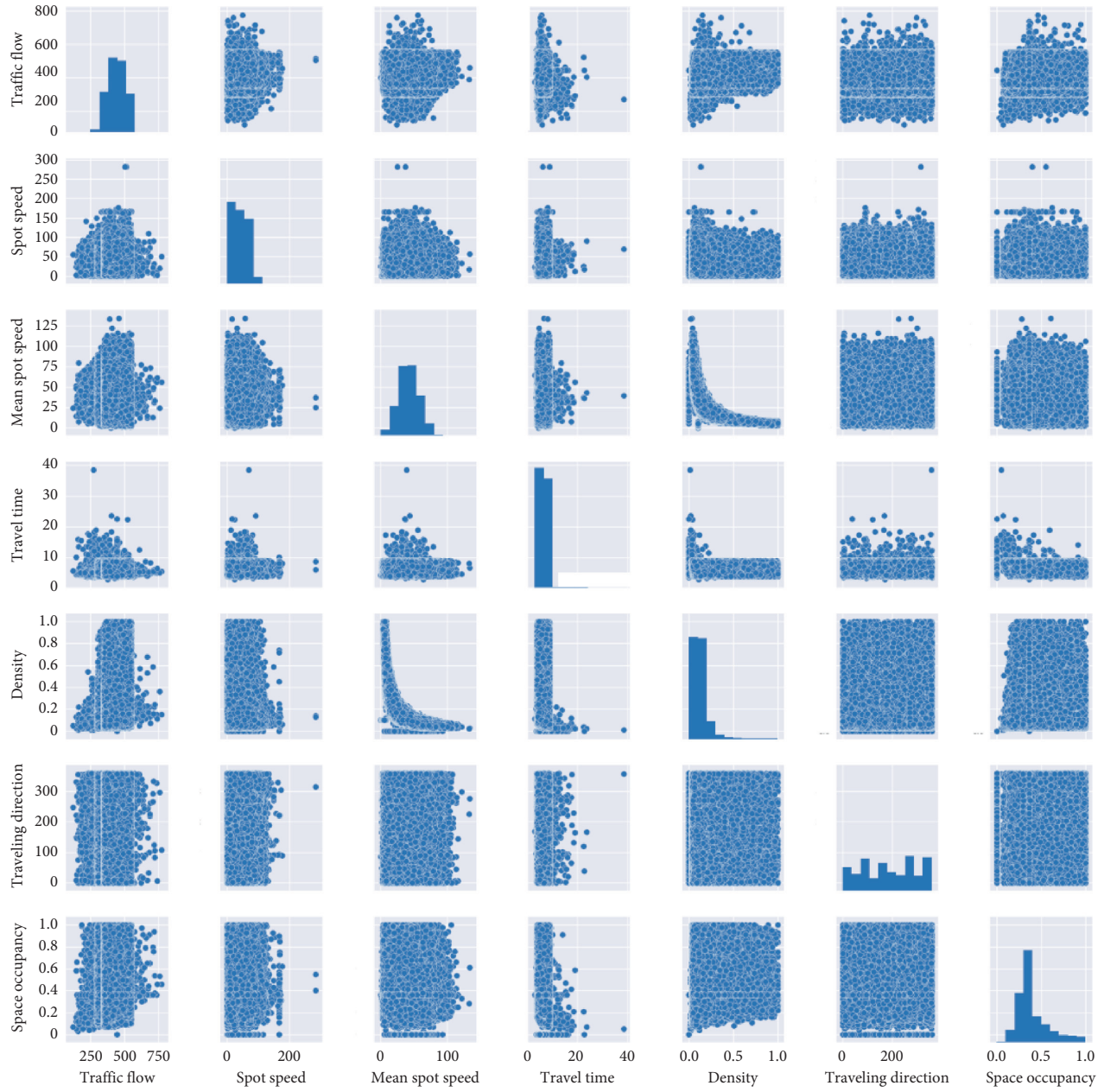

Figure 6: Interfeature relationship exploration map.

different combinations of key features will affect the final prediction results of the model, we choose two different feature input schemes. The first scheme uses all data features as the input to the model, and the prediction evaluation results are shown in Table 5. In the second scheme, XGBoost and LightGBM in Figures 8 and 9 are deleted according to the last two of data feature importance, and the remaining 11 key features are taken as model input for prediction. After prediction, the evaluation results are shown in Table 6.

We can see that, under many indicators, the LightGBM model can obtain better prediction results, and the fusion model combining XGBoost and LightGBM can further improve the previous prediction results.

In addition, we also found that the prediction effect of the three models on all data feature inputs was not ideal, but the prediction accuracy was significantly improved when only the key features were taken as input. We believe that the redundant data features have low volatility and a single value (such as weather conditions, only sunny, cloudy, fog, and rain), which cannot describe the travel time well and in the complex and useless data stack characteristics under the strategy of the integration of too complex, often produced the phenomenon of fitting. In contrast, choosing a large number of key features as the input for model prediction is a more favorable choice, and it is also suitable for dealing with the prediction problem in this paper.

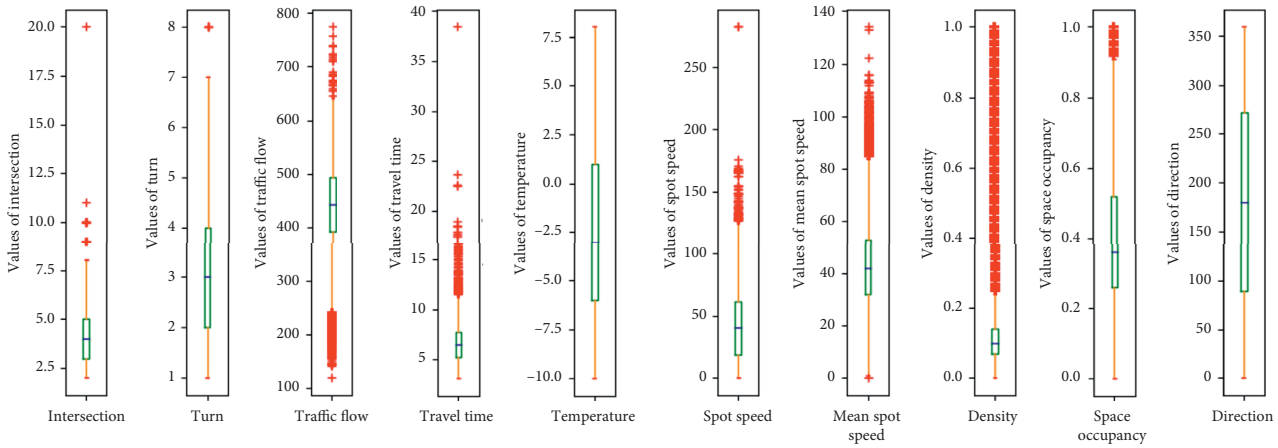

Figure 7: The boxplot of feature values.

Table 4: Example of the training/testing data file.

Spot speed	Mean spot speed	Density	Space occupancy	Traveling direction	Number of intersections	Number of turning	Traffic flow	...
75	53	0.07	0.23	351	4	3	355	...
76	37	0.09	0.62	325	4	3	349	...
38	38	0.14	0.48	344	4	3	487	...
51	42	0.12	0.3	81	4	2	452	...
41	45	0.11	0.3	357	7	5	369	...
57	28	0.17	0.25	154	5	4	421	...
...	...	...	...	...	...	...	...	...

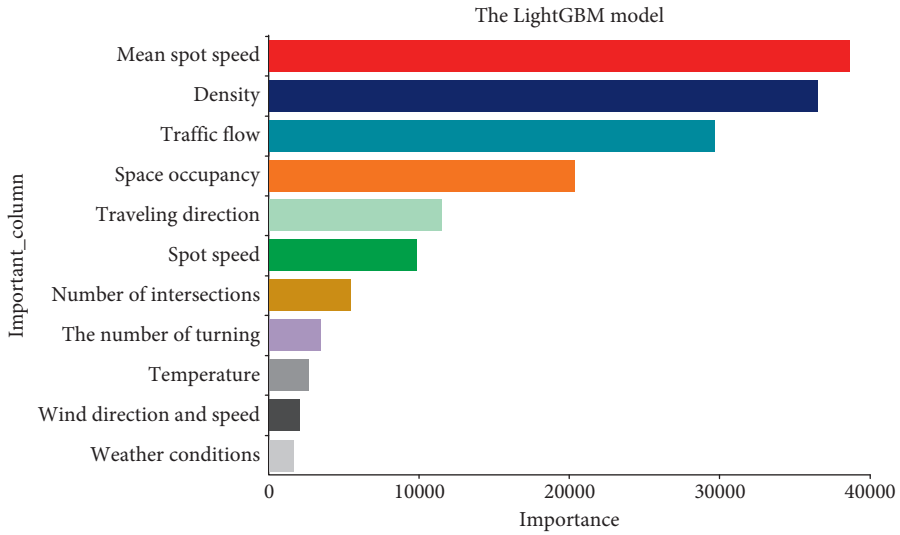

Figure 8: Critical features of LightGBM model. The features are listed in top-down order.

At the same time, in order to verify the effectiveness of the model, this paper introduces the LSTM neural network model for comparative experiments [33]. The data from December 10–18 are used as the training set of the model, and the data from December 19–21 are used as the test set of the model. The data set is divided at 5 min intervals, and the

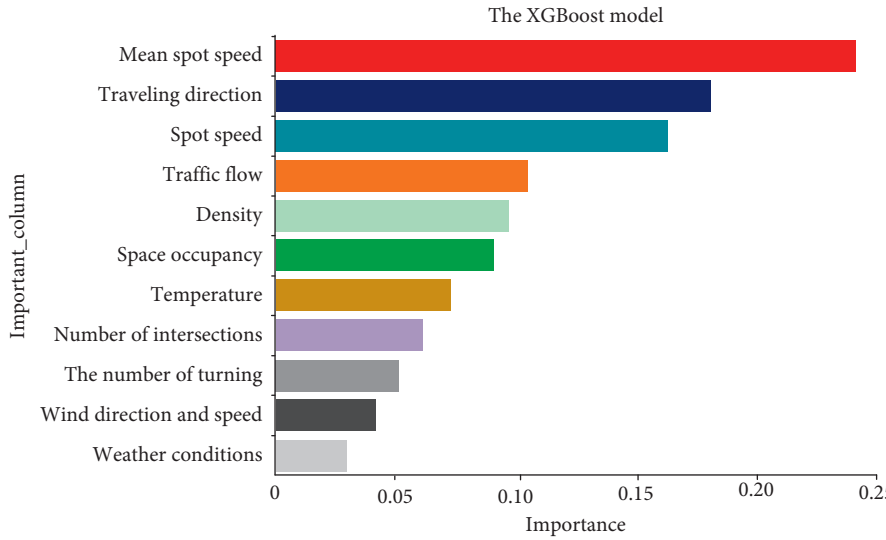

Figure 9: Critical features of XGBoost model. The features are listed in top-down order.

Table 5: Evaluation results with all features as model inputs.

	SMAPE	MAPE	RMSE	MAE
GBDT	3.7695	3.9317	0.4277	0.2483
XGBoost	3.7481	3.9394	0.4451	0.2490
LightGBM	3.3052	3.3305	0.3233	0.2167
Model fusion	3.2651	3.3148	0.3189	0.2144

Table 6: The evaluation results of key features as input to the model.

	SMAPE	MAPE	RMSE	MAE
GBDT	3.7332	3.9118	0.4295	0.2482
XGBoost	3.7280	3.8742	0.4134	0.2452
LightGBM	3.2544	3.2882	0.3203	0.2141
Model fusion	3.2439	3.2851	0.3197	0.2133

Table 7: Comparison of various models.

ID	1	2	3	4	5	6	7	...
Average travel time (min)	5.646	5.103	5.673	5.193	5.294	5.543	4.726	...

Table 8: Comparison of various models.

	SMAPE	MAPE	RMSE	MAE
LSTM	9.7652	11.1553	2.0984	1.2541
XGBoost	3.7280	3.8742	0.4134	0.2452
LightGBM	3.2544	3.2882	0.3203	0.2141
Model fusion	3.2439	3.2851	0.3197	0.2133

average travel time of all vehicles within 5 min is taken as the input of the model, as shown in Table 7. The evaluation results after prediction are shown in Table 8.

Compared with other models, the LSTM model has poor prediction effect. We believe that because only the average travel time is selected as the model input and the multivariable input method is used, the prediction effect may be improved. At the same time, the average travel time selected as input in this paper is relatively macro; if the travel time of a single or partial road segment is predicted, the effect may be better.

5. Conclusion

In this paper, we propose a model fusion-based method to predict the travel time of urban macrovehicles based on multifeature data sets such as vehicle GPS and weather conditions. By mining the original GPS data set and combining it with spatial features, we obtained a more comprehensive data set, and we analyzed the importance of

model features by means of data distribution and box graphs to filter redundant features. The results show that the most important influence variables in the two models are mostly the same. Finally, we weighted fusion the two models of XGBoost and LightGBM to improve the prediction effect.

By analyzing the experimental results, we find that the fusion model proposed in this paper can effectively use the analyzed time and space key features to predict travel time. When the full features and key features are used as inputs, the prediction effect is better than the XGBoost and LightGBM models, so the fusion model has good robustness and can effectively predict vehicle travel time.

Although the proposed fusion model has a good performance in predicting travel time, the training and prediction time consumption is more than that of each single model, so the model efficiency needs to be improved. At the same time, this article has considered road characteristics and weather characteristics but lacks research on driver behavior characteristics. Therefore, our future work will analyze the influence of road characteristics, weather characteristics, driver behavior characteristics, and other characteristics on vehicle travel time.

Data Availability

The implementation and data sets used in this paper are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation China (Grant nos. 51408257 and 51308249), Jilin Special Fund for Industrial Innovation (2019C024), Jilin Science and Technology Development Project (20190101023JH), and Jilin Education Department "13th five-year" Science and Technology Project (JJKH20180153).

References

[1] D. M. Miranda and S. V. Conceição, "The vehicle routing problem with hard time windows and stochastic travel and service time," *Expert Systems with Applications*, vol. 64, pp. 104–116, 2016.

[2] Y. Wang, "A two-stage algorithm for origin-destination matrices estimation considering dynamic dispersion parameter for route choice," *PLoS One*, vol. 11, no. 2, Article ID e0149827, 2016.

[3] W.-H. Lee, S.-S. Tseng, and S.-H. Tsai, "A knowledge based real-time travel time prediction system for urban network," *Expert Systems with Applications*, vol. 36, no. 3, pp. 4239–4247, 2009.

[4] J. W. C. Van Lint, *Reliable Travel Time Prediction for Freeways*, TRAIL Research School, Delft, Netherlands, 2004.

[5] M. Abdollahi, M. Arvan, A. Omidvar, and F. Ameri, "A simulation optimization approach to apply value at risk analysis on the inventory routing problem with backlogged demand," *International Journal of Industrial Engineering Computations*, vol. 5, no. 4, pp. 603–620, 2014.

[6] S. K. Farokhi, M. Hamedi, and A. Haghani, "Evaluating moving average techniques in short-term travel time prediction using an AVI data set," in *Proc. 89th Annu. Meeting Transp. Res. Board*, 2010.

[7] X. Zhang and J. A. Rice, "Short-term travel time prediction," *Transportation Research Part C: Emerging Technologies*, vol. 11, no. 3-4, pp. 187–210, 2003.

[8] C. Nanthawichit, T. Nakatsuji, and H. Suzuki, "Application of probe-vehicle data for real-time traffic-state estimation and short-term travel-time prediction on a freeway," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1855, no. 1, pp. 49–59, 2003.

[9] L. Chu, S. Oh, and W. Recker, "Adaptive Kalman filter based freeway travel time estimation," in *Proc. 84th TRB Annu. Meeting*, pp. 1–21, 2005.

[10] Y. Zhang, R. Sun, A. Haghani, and X. Zeng, "Univariate volatility-based models for improving quality of travel time reliability forecasting," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2365, no. 1, pp. 73–81, 2013.

[11] Y. Kamarianakis and P. Prastacos, "Space-time modeling of traffic flow," *Computers & Geosciences*, vol. 31, no. 2, pp. 119–133, 2005.

[12] 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.

[13] N. Julio, R. Giesen, and P. Lizana, "Real-time prediction of bus travel speeds using traffic shockwaves and machine learning algorithms," *Research in Transportation Economics*, vol. 59, pp. 250–257, 2016.

[14] A. Gal, A. Mandelbaum, F. Schnitzler, A. Senderovich, and M. Weidlich, "Traveling time prediction in scheduled transportation with journey segments," *Information Systems*, vol. 64, pp. 266–280, 2017.

[15] G. Jiang, L. Qi, and D. Shuo, "Travel time estimation method using SCATS traffic data based on K-NN algorithm," *Journal of Southwest Jiaotong University*, vol. 48, no. 2, pp. 343–349, 2013.

[16] Q. Bing, D. Qu, X. Chen, F. Pan, and J. Wei, "Arterial travel time estimation method using SCATS traffic data based on KNN-LSSVR model," *Advances in Mechanical Engineering*, vol. 11, no. 5, 2019.

[17] R. B. Sharmila, N. R. Velaga, and A. Kumar, "SVM-based hybrid approach for corridor-level travel-time estimation," *IET Intelligent Transport Systems*, vol. 13, no. 9, pp. 1429–1439, 2019.

[18] Y. Wei and M.-C. Chen, "Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks," *Transportation Research Part C: Emerging Technologies*, vol. 21, no. 1, pp. 148–162, 2012.

[19] F. Zheng and H. Van Zuylen, "Urban link travel time estimation based on sparse probe vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 31, pp. 145–157, 2013.

[20] Y. Zhang, R. Zhang, Q. Ma et al., "A feature selection and multi-model fusion-based approach of predicting air quality," *ISA Transactions*, 2019.

[21] H. R. Varian, "Big data: new tricks for econometrics," *Journal of Economic Perspectives*, vol. 28, no. 2, pp. 3–28, 2014.

[22] J. O. Ogutu, H.-P. Piepho, and T. Schulz-Streeck, "A comparison of random forests, boosting and support vector machines for genomic selection," *BMC Proceedings*, vol. 5, no. S3, 2011.

[23] H. Lin, "A super-learner model for tumor motion prediction and management in radiation therapy: development and feasibility evaluation," *Scientific Reports*, vol. 9, no. 1, pp. 1–11, 2019.

[24] T. Khaleghi, M. Abdollahi, and A. Murat, "Machine learning and simulation/optimization approaches to improve surgical services in healthcare," *Analytics, Operations, and Strategic Decision Making in the Public Sector*, pp. 138–165, 2019.

[25] J. Elith, J. R. Leathwick, and T. Hastie, "A working guide to boosted regression trees," *Journal of Animal Ecology*, vol. 77, no. 4, pp. 802–813, 2008.

[26] X. Ma, C. Ding, S. Luan, Y. Wang, and Y. Wang, "Prioritizing influential factors for freeway incident clearance time prediction using the gradient boosting decision trees method," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2303–2310, 2017.

[27] X. Li, "Travel time prediction in transport and logistics," *VINE Journal of Information and Knowledge Management Systems*, vol. 49, no. 3, pp. 277–306, 2019.

[28] 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.

[29] Z. He, G. Qi, L. Lu, and Y. Chen, "Network-wide identification of turn-level intersection congestion using only low-frequency probe vehicle data," *Transportation Research Part C: Emerging Technologies*, vol. 108, pp. 320–339, 2019.

[30] L. Lin, Z. He, and S. Peeta, "Predicting station-level hourly demand in a large-scale bike-sharing network: a graph convolutional neural network approach," *Transportation Research Part C: Emerging Technologies*, vol. 97, pp. 258–276, 2018.

[31] L. Liu, "A cohesion-based heuristic feature selection for short-term traffic forecasting," *IEEE Access*, vol. 7, pp. 3383–3389, 2018.

[32] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[33] C. Yang, "Research on urban road travel time prediction method based on deep learning," MS thesis, Southwest Jiaotong University, Chengdu, China, 2019.