International Journal of
Geo-Information

MDPI

Article

Urban Link Travel Time Prediction Based on a Gradient Boosting Method Considering Spatiotemporal Correlations

Faming Zhang [1], Xinyan Zhu [1,2], Tao Hu [2,*], Wei Guo [1,2,*], Chen Chen [3] and Lingjia Liu [1]

[1] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; zhang.fa.ming@163.com (F.Z.); geozxy@263.net (X.Z.); liulingjia_office@163.com (L.L.)
[2] Collaborative Innovation Center of Geospatial Technology, 129 Luoyu Road, Wuhan 430079, China
[3] School of Geodesy and Geomatics, Wuhan University, 129 Luoyu Road, Wuhan 430079, China; cherriechen@whu.edu.cn
* Correspondence: taohu07@hotmail.com (T.H.); guowei-lmars@whu.edu.cn (W.G.)

Academic Editor: Wolfgang Kainz
Received: 9 August 2016; Accepted: 31 October 2016; Published: 4 November 2016

Abstract: The prediction of travel times is challenging because of the sparseness of real-time traffic data and the intrinsic uncertainty of travel on congested urban road networks. We propose a new gradient–boosted regression tree method to accurately predict travel times. This model accounts for spatiotemporal correlations extracted from historical and real-time traffic data for adjacent and target links. This method can deliver high prediction accuracy by combining simple regression trees with poor performance. It corrects the error found in existing models for improved prediction accuracy. Our spatiotemporal gradient–boosted regression tree model was verified in experiments. The training data were obtained from big data reflecting historic traffic conditions collected by probe vehicles in Wuhan from January to May 2014. Real-time data were extracted from 11 weeks of GPS records collected in Wuhan from 5 May 2014 to 20 July 2014. Based on these data, we predicted link travel time for the period from 21 July 2014 to 25 July 2014. Experiments showed that our proposed spatiotemporal gradient–boosted regression tree model obtained better results than gradient boosting, random forest, or autoregressive integrated moving average approaches. Furthermore, these results indicate the advantages of our model for urban link travel time prediction.

Keywords: urban link travel time prediction; spatiotemporal correlations; spatiotemporal gradient–boosted regression tree model; big data

1. Introduction

Estimating and predicting travel times is challenging because of the intrinsic uncertainty of travel on congested urban road networks and uncertainty stemming from the collection of data with probe vehicles equipped with GPS. Uncertainty is produced by fluctuations in traffic and affected by many other factors, such as traffic demand (e.g., due to population characteristics, seasonal effects, time instant, driver behavior, the availability of traffic information, and user responses), traffic control (e.g., due to accidents, road work, and road geometry), weather conditions (e.g., due to temperature, rain, snow, and wind), stochastic arrivals and departures at signalized intersections [1], and the travel direction of traffic flows. These random fluctuations are often complicated and difficult to predict. Understanding these fluctuations is especially necessary when developing more accurate prediction algorithms. Meanwhile, due to the low frequency [2–4] of probe vehicle GPS data acquisition and the regional limitation of driving areas, trajectory information collected by probe vehicle GPSs cannot cover an entire urban road network. Therefore, the collected data are sparse [5,6]. Estimating and predicting link travel time using sparse data is a challenge that must be solved for accurate estimation and prediction of travel times.

Corresponding to the needs of travel time prediction, many prediction methods have been proposed, including statistical and regression methods [7–9], historical average and smoothing [10–12], diverse machine learning [13,14], and traffic flow theory-based methods [15]. Among these methods, the Autoregressive Integrated Moving Average Model (ARIMA) model is gradually becoming a benchmark for evaluation of newly developed prediction models [16]. The ARIMA model [7,17] generally assumes a certain model structure for the data and provides interpretable parameters with a simple model structure. This model can better predict travel time when traffic flow exhibits patterns of regular change. Another effective prediction method involves machine learning algorithms, which are also widely applied in traffic prediction. Successful applications include support vector machines (SVM) [13,18], neural networks [14,19] and hybrid and ensemble techniques [13,20]. In contrast to existing statistical models, in machine learning, it is not necessary to assume that the data have a certain structure; this structure can be unknown. Machine learning algorithms can capture the potential model structure of data [21]. An important disadvantage of this approach, however, is the lack of interpretability that limits the application of this model.

In recent years, ensemble algorithms have become important for solving prediction and classification problems in many different fields with certain achievements [22]. Among all the ensemble algorithms, tree-based ensemble algorithms are one of the most important methods. Instead of fitting a single model, tree-based methods combine multiple single tree models to obtain optimal prediction performance. This approach produces better predictions and may help policy makers better understand the relationship between traffic and the factors that impact it. Moreover, tree-based ensemble algorithms require less data preprocessing and provide better fits to nonlinear relationships. These advantages make the tree-based approach a good choice when addressing traffic analysis.

There is limited research, however, on the use of tree-based algorithms in the transportation field. Hamner [23] applied the random forest algorithm to predict travel time and showed that the proposed model outperformed other models in terms of prediction precision. Wang [24] applied an ensemble bagging decision tree to forecast the influence of weather on airport capacity and demonstrated that its performance is better than that of the SVM algorithm. Ahmed and Abdel-Aty [25] identified transportation risks using data obtained from different sensors; the results showed that the stochastic gradient boosting method is superior to traditional statistical methods. Similarly, Chung [26] applied a gradient regression tree to study crash occurrences. These latter two studies utilized a boosting algorithm to address classification and prediction problems, rather than travel time prediction. Yanru Zhang [27] utilized a gradient boosting method to improve travel time prediction considering real travel time but ignored information from historical travel time data and the spatiotemporal correlation between target and adjacent links. In addition, this approach cannot efficiently predict link travel time under sparse data conditions. The existing research illustrates the effectiveness and efficiency of tree-based algorithms. Nevertheless, there is little research on the use of gradient boosting trees to predict travel time.

To fill this gap, our research presents a tree-based ensemble algorithm to predict urban link travel time considering relevant input variables derived from historical travel time and real travel time. At the same time, we consider the spatiotemporal correlation between target and adjacent links when calculating urban link travel time. Our proposed algorithm exploits the Spatiotemporal Gradient–boosted regression tree (STGBRT) model from machine learning to predict link travel time. The STGBRT model uncovers underlying patterns in travel time data to enhance the accuracy and interpretability of the model. In contrast to other tree-based models, the gradient boosting tree approach assigns a lower weight to trees that produce incorrect classifications generated by the regression tree model while identifying an optimal combination of trees. The gradient boosting method has the potential to provide more accurate predictions than random forest algorithms.

The article is outlined as follows. In Section 2, a detailed description of single regression tree and gradient–boosted regression tree methods is provided. In Section 3, the standardization of measurement and correlation between the target and adjacent links is described. In Section 4, we describe our experiment, including the data we used, the application of our model, and the comparison of our model to others. A discussion of the results and some conclusions are outlined at the end.

2. Methodology

Ensemble algorithms based on multiple basic models, such as neural networks, random forests, decision trees, and k-Nearest Neighbors, can obtain higher accuracy in estimation and prediction. In an ensemble algorithm, every basic model can provide a solution to a problem. These predictions are combined in some way, such as weighting or averaging, to generate a final output. In general, the prediction accuracy of an ensemble model is superior to that of the basic models included in the ensemble model [28]. The predictions of ensemble models can be understood from the following example. For instance, we usually ask for other people's opinions when we make decisions. Each person will give a solution to the problem based on their own experience. We can make a more accurate decision by comprehensively measuring all the opinions. Ensemble algorithms reduce decision-making errors by correcting mistakes in each basic model.

Of the possible base models, decision trees, also called regression trees, are among the most commonly used approaches. In operations research, decision trees help identify a strategy to reach a goal, and they are also a popular tool in machine learning. A decision tree is a flowchart-like structure in which each internal node represents a "test" performed on an attribute (e.g., whether a coin flip comes up heads or tails); each branch represents the outcome of the test and each leaf node represents a class label. The paths from root to leaf represent classification rules. Decision tree algorithms have many attractive properties, such as low training time and complexity, fast prediction processing, and straightforward demonstration. At the same time, they have disadvantages, such as overfitting. Tree-based ensemble algorithms establish many individual trees, combining the results of each tree for more accurate results. In general, there are two types of ensemble algorithms based on trees, the random forest method and the gradient–boosted regression tree algorithm [29]. A single regression tree is used as the base model in these two algorithms. Section 2.1 briefly explains how single regression trees work and illustrates the process of constructing a gradient–boosted regression tree (GBRT).

2.1. Single Regression Tree

As with all regression techniques, we assume the existence of a single output variable (response) and one or more input variables. The general regression tree-building methodology allows input variables to be a mixture of continuous and categorical variables. A regression tree may be considered a variant of decision trees, which are designed to approximate real-valued functions instead of being used for classification tasks. A regression tree is built through a process known as binary recursive partitioning [30]. This is an iterative process of splitting the data into partitions and then further splitting the partitions in each of the branches. Initially, all of the records in a training set are together in a single group. The algorithm then tries to divide up the data using every possible binary split on every field. The algorithm chooses the split that partitions the data into two parts such that it minimizes the sum of the squared deviations from the mean in the separate parts. This splitting or partitioning is then applied to each of the new branches. The process continues until each node reaches a user-specified minimum node size and becomes a terminal node.

A single regression tree [27] can be described as follows. As depicted in Figure 1a, the left panel is split into five regions, {R1, R2, R3, R4, and R5}, according to two variables X1 and X2 using four split points b1, b2, b3, and b4. The size of the regression tree in Figure 1 is the total number of end nodes because the tree was partitioned into five different regions, which is equal to the number of end nodes of the tree. The right panel of Figure 1 is a binary tree representation of the same model, expressing five different split regions.

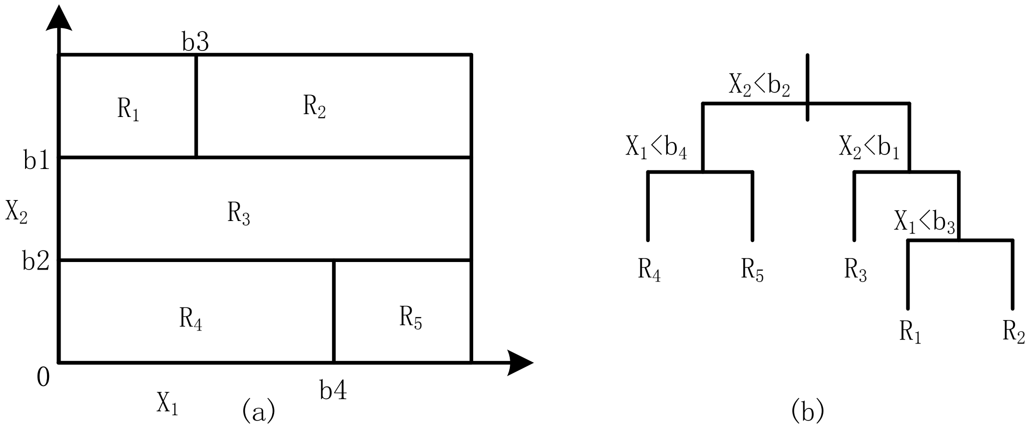


Figure 1. Single regression tree.

Now, we consider a general question of the same type as the example shown in Figure 1, which includes p inputs with one output corresponding to the input of the regression problem. For example, we have n observations and each observation is denoted as $y_i, x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}, \ldots, x_{ip}$ for $i = 1, 2, \ldots, n$. For travel time prediction, y_i is the dependent variable and is regarded as the predicted travel time corresponding to the ith observation. $x_{i1}, x_{i2}, x_{i3}, \ldots, x_{ij}, \ldots, x_{ip}$ are independent variables relevant to the prediction of travel time, such as historical travel time, real-time travel time, traffic volume, time instant, and weather or other external factors. Let us assume that the feature space is divided into m regions $R_1, R_2, \ldots, Rm$ representing the different regions of different traffic conditions. Thus, the traffic state is divided into different categories by an input parameter, and the corresponding model is established for each type of dependent variable. Generally, the expected value in each region of the dependent variable is regarded as a constant C_m. It is an expected optimal value that we hope to obtain using the independent variables. If the optimality criterion is to minimize the sum of squares of the deviation, then the optimal value of C_m is the average of the y_i values in the area of R_m [31]. As shown in Figure 1a, we estimated different values in the area R_m. In this research, we use a greedy algorithm [32,33] to determine the best split variables and split points. The single regression tree is the basic model for the gradient–boosted regression tree.

2.2. Gradient–Boosted Regression Tree

The idea of gradient boosting originated from the observation made by Leo Breiman [34] that boosting can be interpreted as an optimization algorithm on a suitable cost function. Explicit gradient boosting regression algorithms were subsequently developed by Jerome H. Friedman [35,36]. Mason et al. [37] introduced the abstract view of boosting algorithms as iterative functional gradient descent algorithms; that is, they are algorithms that optimize a cost function over function space by iteratively choosing a function (a weak hypothesis) pointing down the gradient. This functional gradient view of boosting has led to the development of boosting algorithms in many areas of machine learning and statistics beyond regression and classification. Gradient Tree Boosting, also termed the Gradient–Boosted Regression Tree (GBRT) method, is a generalization of boosting applied to arbitrary differentiable loss functions. Gradient boosting is a machine learning technique for regression and classification problems that produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stepwise fashion, similar to other boosting methods, and generalizes these methods by allowing optimization of an arbitrary differentiable loss function.

Friedman [35] put forward an improvement to the method of gradient boosting using fixed size regression trees as the basic model. The modified model improves the quality of the gradient boosting model [37]. In this study, an improved gradient-boosted regression tree model, the spatiotemporal gradient–boosted regression tree (STGBRT) model, is proposed for travel time prediction. This model considers spatiotemporal correlations between target and adjacent links. Assuming that the number of leaves for each tree is J, the space of the m-*th* tree can be divided into J disjoint subspaces, such as R_{1m}, R_{2m}, . . . , R_{Jm}, and the predicted value for subspace R_{Jm}, is the constant b_{jm}. Therefore, the regression tree can be expressed by Equations (1) and (2):

$$g_m\left(x_i\right) = \sum_{\mathrm{j}=1}^{J} b_{jm} I\left(x_i \in R_{jm}\right) \tag{1}$$

$$I\left(x_i \in R_{jm}\right) = \begin{cases} 1,\ if\ x_i \in R_{jm} \\ 0,\ otherwise \end{cases} \tag{2}$$

To minimize the loss function of the STGBRT model, we use the steepest descent method, which is among the simplest frequently used numerical minimization methods Following the numerical optimization paradigm, we take the approximate solution, $F\left(x_i\right)$, to be

$$F\left(x_i\right) = \sum_{m=0}^{M} f_m\left(x_i\right) \tag{3}$$

where $f_0\left(x\right)$ is an initial guess, M denotes the index of the tree, and $\left\{f_m\left(x_i\right)\right\}_1^M$ are incremental functions defined by the optimization method [35]. Using the steepest descent method, there exists the following equation

$$f_m\left(x_i\right) = -\rho_m g_m\left(x_i\right) \tag{4}$$

The current gradient g_m, is computed according to Equation (5) [35], based on the sequence of preceding steps. It defines an increment. In Equation (5), $f\left(x_i\right)$ is an estimation or approximation of observation y_i that corresponds to "input" or "explanatory" variables, $\mathbf{x} = \left\{x_1, \ldots, x_n\right\}$,

$$g_m\left(x_i\right) = \left[\frac{\partial L\left(y_i, f\left(x_i\right)\right)}{\partial f\left(x_i\right)}\right]_{f(x_i)=f_{m-1}(x_i)} \tag{5}$$

The multiplier ρ_m in Equation (4) is given according to Equation (6):

$$\rho_m = \mathrm{argmin}_\rho \sum_{i=1}^{n} L(y_i, f_{m-1}\left(x_i\right) + \rho_m g_m\left(x_i\right)) \tag{6}$$

The model is updated according to Equation (7):

$$F_m\left(x_i\right) = F_{m-1}\left(x_i\right) + \rho_m g_m\left(x_i\right) \tag{7}$$

The gradient–boosted regression tree method establishes a new model in the direction of residual decrease and updates the model by minimizing the expectations of the loss function according to Equations (5)–(7). This step is the most important part of gradient boosting. In general, the fitted model can reduce its training error with an increase in the number of basic trees in the model. However, it will also reduce the generalizing ability of the fitted model if the model is too close to the training data. By increasing the number of iterations, the model becomes complex, so minor fluctuations in the data are exaggerated. This added complexity will cause poor prediction performance for the test data. Consequently, it is necessary to determine the optimal number of iterations for the model to minimize potential prediction errors. The overfitting phenomenon can also be avoided by controlling the number of iterations, the number of basic trees, and the learning rate. The STGBRT model strategically makes each basic model achieve minimum loss. It uses a stage-wise sampling strategy, which pays more attention to unfavorable examples. This feature distinguishes it from the random forest model that trains each model using random sampling or equal probability sampling. Therefore, the performance of the STGBRT model is influenced by the number of trees and the learning rate. The optimal performance of the model can be obtained by carefully selecting the best combination of these parameters [38].

3. Measurement and Correlation in Space and Time

3.1. Spatial Correlation

Many indices have been designed to quantitatively measure the correlations among spatial and temporal data, and most of these indices are based on Pearson's coefficient [39]. In statistics, the Pearson correlation coefficient (referred to as PCC or Pearson) is a measure of the linear correlation between two variables X and Y and takes a value between −1 and +1. If the value is 1, it indicates a perfect positive correlation; while 0 indicates no correlation and −1 indicates perfect negative correlation. It is widely used in the sciences as a measure of the degree of linear dependence between two variables and was developed by Karl Pearson. Given two variables X and Y, the Pearson's correlation coefficient is defined as follows:

$$\rho_{X,Y} = \frac{E\left[\left(X-\mu_X\right)\left(Y-\mu_Y\right)\right]}{\sigma_X \sigma_Y} \tag{8}$$

where μ_X and μ_Y are the averages of variables X and Y, respectively. Similarly, σ_X and σ_Y are the corresponding standard deviations of variables X and Y. The spatial correlation coefficient between a target link and an adjacent link can be calculated according to Equation (8).

The schematic diagram in Section 4.1 reveals the traffic flow, where link 82 is a target link, link 88 is an upstream link, and link 77 is a downstream link. In this research, the time step was set to 30 min. Therefore, we extracted the expected speed associated with corresponding links in a certain direction every 30 min. Table 1 shows the pairwise correlations between individual links on a subset of the network according to Equation (8). As can be inferred from Table 1, the correlation coefficient of the expected speed in a certain direction and for a different time between every two links for links 82, 77, and 88 are significantly correlated at the 0.01 confidence level (two-tailed). The correlation coefficients for speed on different days have different values and vary by the day. Figure 2 is a line chart reflecting the relationship of expected speed between links 77, 82, and the adjacent link 88 from Monday to Friday. As seen in the line chart, the expected speed for link 82 increases when the expected speed of the adjacent link 88 increases, indicating a positive correlation. It is also seen in Figure 2 that the expected speed of links 77, 82, and 88 has a rhythmic pattern. Consequently, both Table 1 and Figure 2 reflect the dynamic spatial correlations between a target link and an adjacent link. Thus, we selected adjacent link information as model inputs for target link travel time prediction.

Table 1. The correlation coefficient of expected speed in a certain direction at different times among target link 82, adjacent link 77, and adjacent link 88.

Link	Monday	Tuesday	Wednesday	Thursday	Friday
Link 77, link 82 in −1 traffic flow direction	0.755327 **	0.599857 **	0.451914 **	0.575618 **	0.558733 **
Link 88, link 82 in −1 traffic flow direction	0.719256 **	0.837093 **	0.762925 **	0.715509 **	0.605603 **

** Significantly correlated at the 0.01 confidence level (two-tailed). * Significantly correlated at the 0.05 confidence level (two-tailed).

3.2. Temporal Correlation

The temporal autocorrelation function (TACF) [40] treats two time series as a bivariate stochastic process and measures the covariance coefficients between each series at specified lags. For example, if there is a time series at time t for variable X, then there exists another time series at lag time k

corresponding to variable X at time *t-k*. Then, the correlation coefficient of these two time series corresponding to X can be denoted as in the following equation:

$$\rho_k = \frac{E\left[\left(X_t - \mu\right)\left(X_{t-k} - \mu\right)\right]}{{\sigma_X}^2} \tag{9}$$

where μ is the mean of variable X and σ_X is the corresponding standard deviation of variable X.

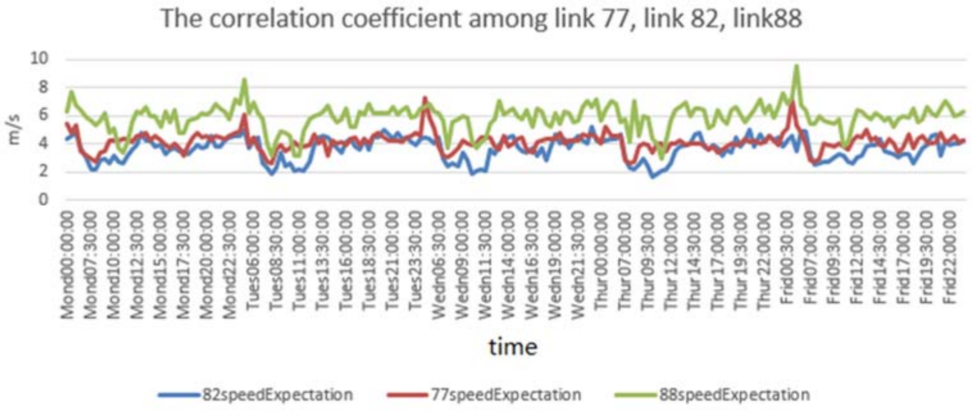


Figure 2. The correlation coefficient among link 77, link 82, and link 88.

In fact, a temporal autocorrelation coefficient can be measured simply by taking the correlation of a variable with a lagged specification of itself. Therefore, the temporal autocorrelation was measured by modifying PCC to include this lagged specification. The temporal difference of variable X is measured between time t and time *t*–*k* according to Equation (9). If the process is stationary, then ${\sigma_X}^2$ can be used as the deviation of x and is assumed to be constant over time. Table 2 reveals the temporal autocorrelation of link 82 at different lag times corresponding to time t.

Table 2. Temporal autocorrelation of link 82 for different lag times relative to a particular time t.

	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9
t	1	0.774 **	0.557 **	0.365 *	0.224	0.169	0.189	0.114 **	−0.014 **	−0.104 *
t+1	0.774 **	1	0.741 **	0.542 **	0.377 **	0.236	0.168	0.202	0.139	0.040
t+2	0.557 **	0.741 **	1	0.734 **	0.516 **	0.363 *	0.225	0.128	0.201	0.142
t+3	0.365 *	0.542 **	0.734 **	1	0.724 **	0.511 **	0.358 *	0.205	0.132	0.211
t+4	0.224	0.377 **	0.516 **	0.724 **	1	0.727 **	0.508 **	0.350 *	0.223	0.169
t+5	0.169	0.236	0.363 *	0.511 **	0.727 **	1	0.725 **	0.511 **	0.360 *	0.244
t+6	0.189	0.168	0.225	0.358 *	0.508 **	0.725 **	1	0.725 **	0.514 **	0.366 *
t+7	0.114	0.202	0.128	0.205	0.350 *	0.511 **	0.725 **	1	0.749 **	0.554 **
t+8	−0.014	0.139	0.201	0.132	0.223	0.360 *	0.514 **	0.749 **	1	0.753 **
t+9	−0.104	0.040	0.142	0.211	0.169	0.244	0.366 *	0.554 **	0.753 **	1

** Significantly correlated at the 0.01 confidence level (two-tailed). * Significantly correlated at the 0.05 confidence level (two-tailed).

4. The Experiment

In contrast to estimation methods, the purpose of travel time prediction is to forecast the travel time for a trajectory that will start at a particular moment, using historical and current travel time for that trajectory. A prediction is made now or in the future [41]. For this purpose, traffic data of target and adjacent links from past and current data were used as depicted in Figure 3, which shows a schematic diagram of travel time prediction based on past data combined with current data. Therefore, both real-time traffic data and big data reflecting historical traffic conditions contribute to link travel time prediction. Real-time traffic data more accurately reflect current traffic states.

Travel time prediction models the correlation of different variables with available traffic information. Consequently, the more comprehensive the information we extract is, the more accurate the travel time prediction results will be. Considering that traffic characteristics are a complex phenomenon involving non-linear and chaotic characteristics, it is often difficult to construct an exact equation expressing the relationship among different characteristics. Data-driven approaches are a promising area in modeling and predicting traffic. The following subsections discuss in detail the application of the spatiotemporal gradient–boosted regression tree model (STGBRT) to travel time prediction.

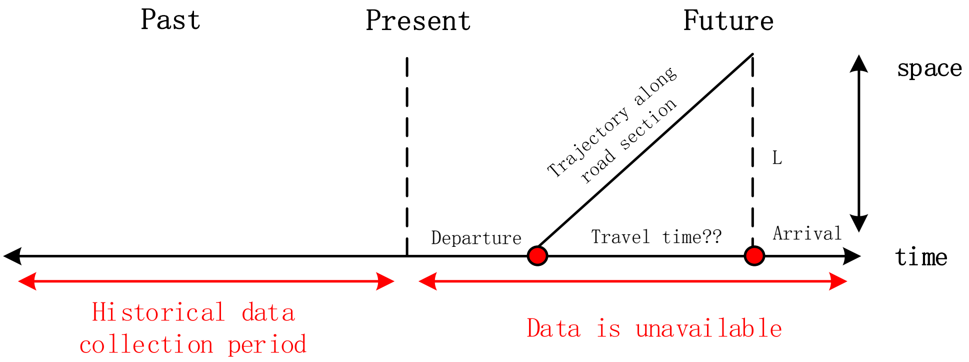


Figure 3. Schematic diagram of travel time prediction.

4.1. Data Description and Preparation

Probe vehicles equipped with GPS as mobile traffic sensors are used to collect network-wide traffic data. In our research, historical and real-time probe vehicle data provided by a private-sector company are used. The Oracle database containing the probe vehicle data were acquired from the Intelligent Transportation System (ITS) in Wuhan, China. Probe vehicles collect information such as instantaneous speeds, timestamps, longitude and latitude coordinates, and compass headings; reflecting the running state of the urban traffic, which plays an important role in real-time or near real-time travel time estimation and prediction. Our research utilized travel time data from probe vehicles operating on the local road network of the city of Wuhan to predict travel time. Table 3 shows location information for selected local roads in Wuhan's road network. These data include section number, starting geographic coordinates, ending geographic coordinates, and the length of each segment. Figure 4 shows the local roads in the Wuhan road network.

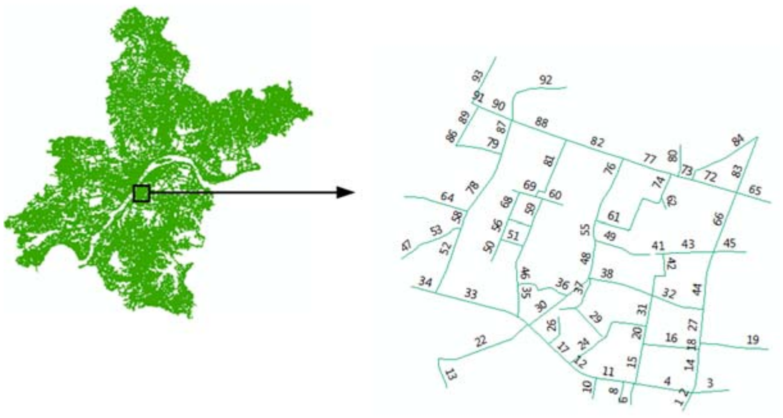


Figure 4. Visualization of the local road network in Wuhan City, China.

Table 3. Selected arterial road network used in the experiment.

Section ID	Start Coordinate		End Coordinate		Length (Meter)
	Latitude	Longitude	Latitude	Longitude	
88	30.535	114.329	30.533	114.334	475.69
82	30.533	114.334	30.532	114.338	489.10
77	30.532	114.338	30.530	114.342	411.43

Due to the effect of GPS positioning error [42], GPS points tend to deviate from the actual road of probe vehicle travel. Therefore, GPS points that deviate from the road network must be first projected to the road according to the probe vehicle passing trajectory. The link travel time of a single probe vehicle is then calculated using these map-matched points. In our research, probe vehicle trajectories were adjusted using a map matching algorithm [43–46]. We calculated travel times and average speeds of probe vehicles traversing target links, taking into consideration the probe vehicles' states at intersections [47–49]. We extracted the characteristics of the links from the massive quantities of statistical travel time data collected by probe vehicles traversing the target link. The obtained statistical data includes link ID, entering endpoint ID, exiting endpoint ID, probe vehicle ID, the moment a probe vehicle entered the link, the travel time for a probe vehicle traversing the link, and the average speed of a probe vehicle traversing the link, as depicted in Table 4. Existing research has shown that probe vehicle trajectories display similar traffic patterns over a weekly cycle [50–53]. Therefore, we extracted characteristics between target and upstream links as historical characteristics according to the weekly cycle. Meanwhile, the accessed data were aggregated over 30-minute time intervals because of the scarcity of travel information. Therefore, one day was divided into 48 time intervals and input characteristics were extracted from this information to predict future travel times. Figure 5 is a schematic diagram of traffic flow on a partial road network from Figure 4 that includes road numbers and traffic direction. In our research, we use our model to predict travel times for link 82 using observed spatiotemporal correlations among link 82, link 88, and link 77. We extracted the spatiotemporal correlation characteristics from big data reflecting historic traffic conditions collected by probe vehicles from January to May, 2014. Next, the eleven weeks of data covering the period from 5 May 2014 to 20 July 2014 were taken as training data for the STGBRT model. Finally, one week of data from 21 July 2014 (Monday) to 25 July 2014 (Friday) was taken as test data to verify the validity of the model.

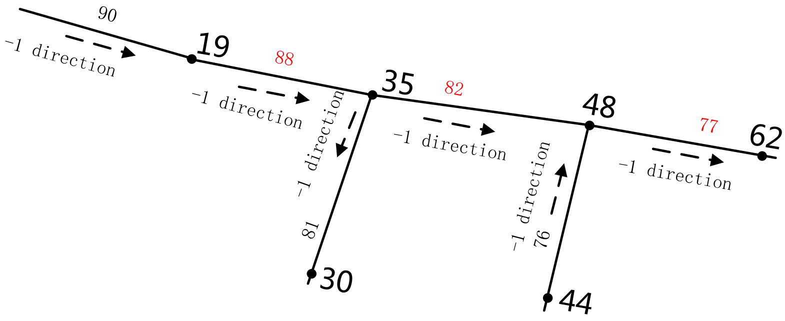


Figure 5. Schematic diagram of traffic flow.

Tables 5–7 summarize travel time information from probe vehicles traveling in the same direction from January to May 2014 in terms of descriptive statistics, including the mean; the standard deviation (SD); the 25th, 50th and 75th percentiles; and the minimum (Min) and maximum (Max) observations. Travel time data were recorded in seconds. From these three tables, it can be inferred that the quartiles of speed for the same link are similar for each day, and differences from day to day are small. In contrast, a great difference in speeds exists among different links. Figures 6–8 show the distributions of observed speeds from link 88, link 82, and link 77, respectively, on Mondays and Wednesdays. Two histograms from the same link show similar patterns, with an approximately normal distribution, if abnormal values are ignored. However, the distributions of travel speed show slight differences among the different links.

Table 4. Travel information from individual probe vehicles.

Link ID	Enter Endpoint ID	Exit Endpoint ID	Probe Vehicle ID	Time Instant	Travel Time (s)	Average Speed (m/s)
82	35	48	23501	2014–06–03 03:17:11	100.0	4.89
82	35	48	22608	2014–06–02 00:00:50	85.0	5.75
82	48	35	29444	2014–06–02 00:12:03	101.0	4.84

Table 5. Basic statistics of travel speed (m/s) for link 88.

Workday	Mean	SD	25th	50th	75th	Min	Max
Monday	6.55	2.22	5.23	6.61	7.8	1.03	26.43
Tuesday	6.56	2.19	5.17	6.61	7.8	1.15	15.35
Wednesday	6.64	1.95	5.41	6.61	7.8	1.46	16.4
Thursday	6.68	2.23	5.52	6.7	7.8	1.43	18.3
Friday	6.42	2.10	5.23	6.34	7.55	1.2	15.86

Table 6. Basic statistics of travel speed (m/s) about link 82.

Workday	Mean	SD	25th	50th	75th	Min	Max
Monday	4.83	2.15	3.27	4.33	6.04	0.94	15.78
Tuesday	4.82	2.08	3.26	4.41	6.09	1.06	13.97
Wednesday	4.73	2.09	3.12	4.25	6.19	1.04	13.59
Thursday	4.77	2.18	3.14	4.33	6.25	0.93	13.22
Friday	4.97	2.20	3.37	4.61	6.04	1.11	17.47

Table 7. Basic statistics of travel speed (m/s) about link 77.

Workday	Mean	SD	25th	50th	75th	Min	Max
Monday	4.42	1.77	3.27	4.16	5.08	1.2	16.46
Tuesday	4.16	1.61	3.21	3.92	4.84	1.12	13.72
Wednesday	4.61	1.81	3.37	4.29	5.14	1.32	13.72
Thursday	4.18	1.52	3.14	4.03	4.84	1.04	11.76
Friday	4.47	1.82	3.37	4.24	5.02	1.24	18.7

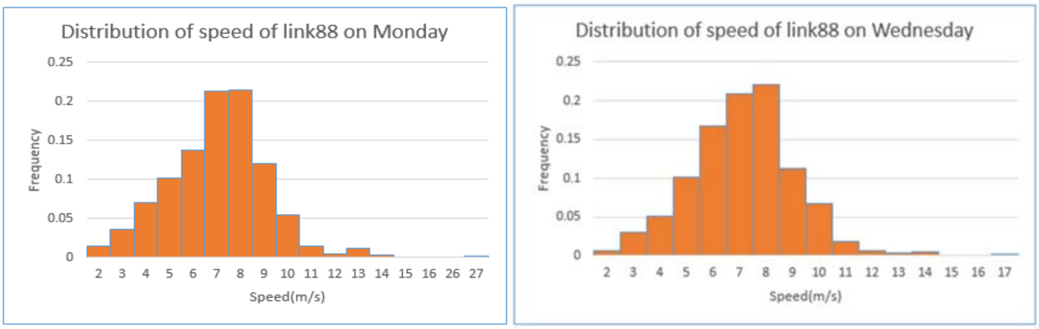


Figure 6. Distributions of speeds observed along link 88 on Monday and Wednesday, respectively.

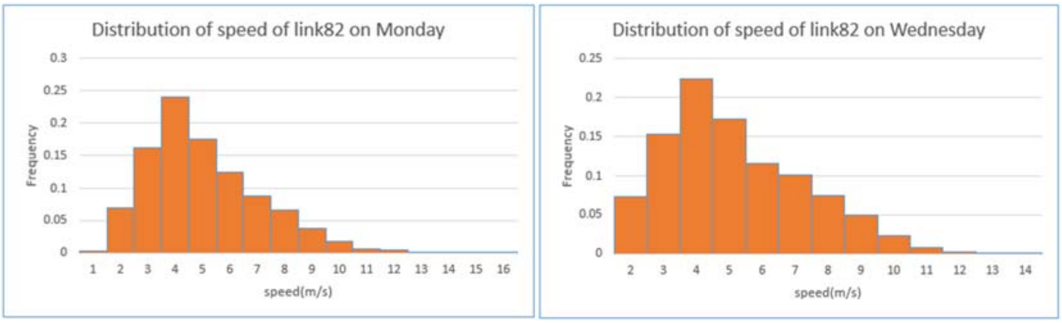


Figure 7. Distributions of speeds observed along link 82 on Monday and Wednesday, respectively.

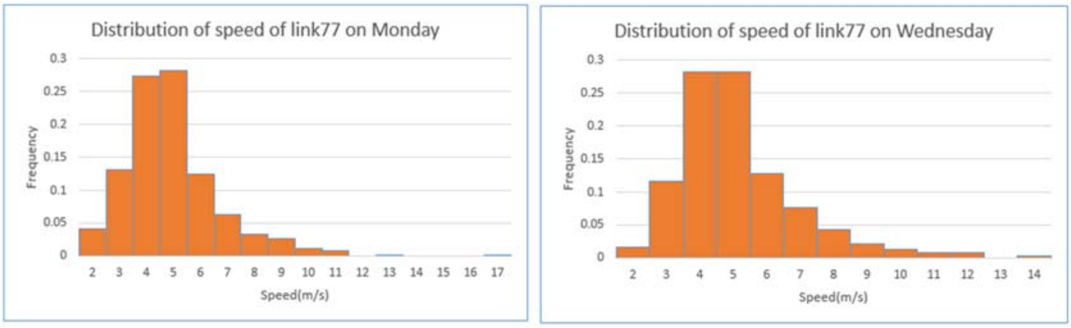


Figure 8. Distributions of speeds observed along link 77 on Monday and Wednesday, respectively.

As shown in Table 2, due to the time distance at the current moment, the temporal autocorrelation is significant within a three-step time period at a 0.01 confidence level when a two-sided test is used. The autocorrelation coefficient decreases with increasing lag time; there is no correlation for lags greater than three time steps. Consequently, it is unnecessary to examine temporal autocorrelation outside a three time step period. Therefore, we selected information collected within two time steps prior to the current time as model inputs when making travel time predictions. We selected several spatiotemporal variables that are relevant to travel time as the inputs and the output for our model, as shown in Table 8. The first 17 columns are the input variables of the model, and the last column is the output variable, which is a function of the inputs. The output of the model is real travel time at lag time t denoted as $tarRTT_t$, and the 17 input variables that were used to predict travel time at lag time t are as follows: weekday, time of day, $tarHTT_{t-1}$, $tarHTT_{t-2}$, $\Delta tarHTT_{\ t-1}$, $tarRTT_{t-1}$, $tarRTT_{t-2}$, $\Delta tarRTT_{\ t-1}$, $UpHTT_{t-1}$, $UpHTT_{t-2}$, $UpRTT_{t-1}$, $UpRTT_{t-2}$, $DoHTT_{t-1}$, $DoHTT_{t-2}$, $\Delta UpHTT_{t-1}$, $\Delta UpRTT_{\ t-1}$ and $\Delta DoHTT_{\ t-1}$. Weekdays are indexed from one to five, representing Monday to Friday; time of day is represented by 30-minute time steps, indexed from 1 to 48. $tarHTT_{t-1}$ and $tarHTT_{t-2}$ are the two most recent historical travel time observations of a target link at times $t-1$ and $t-2$; $\Delta tarHTT_{\ t-1}$ is the growth rate of the historical travel time for a link over two consecutive times, as calculated according to Equation (10). $tarRTT_{t-1}$ and $tarRTT_{t-2}$ are the two most recent real travel time observations for a target link at times t-1 and t-2; $\Delta tarRTT_{\ t-1}$ is the growth rate of real travel time for a link between two consecutive time steps and calculated according to Equation (11). Correspondingly, $UpHTT_{t-1}$ and $UpHTT_{t-2}$ are the two most recent historical travel time observations of upstream links at times $t-1$ and $t-2$; $UpRTT_{t-1}$ and $UpRTT_{t-2}$ are the two most recent real travel time observations of an upstream link at times $t-1$ and $t-2$. $DoHTT_{t-1}$ and $DoHTT_{t-2}$ are the two most recent historical travel time observations of a downstream link at times $t-1$ and $t-2$. Similarly, $\Delta UpHTT_{\ t-1}$, $\Delta UpRTT_{\ t-1}$, and $\Delta DoHTT_{\ t-1}$ are the growth rates of historical travel times for an upstream link, the real travel time growth rate for an upstream link and the historical travel time growth rate for a downstream link between two consecutive time steps, respectively. These variables were calculated according to Equations (12–14). Due to the low frequency of probe vehicle GPS data acquisition and the regional limitations of driving areas, however, the trajectory information collected by a probe vehicle GPS unit cannot cover the entire urban road network; therefore, the data collected are sparse [5,6]. Through analysis, we found that our data lacked sufficient probe vehicle travel information during the period between midnight and 5 a.m. In contrast, data for other time periods were relatively abundant. Therefore, we choose daily traffic data for the period from 6 a.m. and midnight everyday as the research time period. For the missing real-time data during some experimental time intervals, we used travel information from big data reflecting historic traffic conditions corresponding to the time period to make up the missing real-time data.

$$\Delta tarHTT_{t-1} = tarHTT_{t-1} - tarHTT_{t-2} \tag{9}$$

$$\Delta tarRTT_{t-1} = tarRTT_{t-1} - tarRTT_{t-2} \tag{10}$$

$$\Delta UpHTT_{t-1} = UpHTT_{t-1} - UpHTT_{t-2} \tag{11}$$

$$\Delta UpRTT_{t-1} = UpRTT_{t-1} - UpRTT_{t-2} \tag{12}$$

$$\Delta DoHTT_{t-1} = DoHTT_{t-1} - DoHTT_{t-2} \tag{13}$$

4.2. Model Application

To obtain the optimal model, understanding the influence of different parameter combinations on the model performance is critical. Considering input information, we obtained optimal combination parameters of the model to achievelower prediction error. This section shows how performance varies for different choices of parameters. These included the number of trees N and the learning rate *lr* that were used when extracting spatiotemporal characteristics from five months of travel time information collected between January and May 2014. We extracted weekday data from Monday to Friday for the period 5 May to 20 July 2014 as training data. The following five days of data gathered from Monday, 21 July 2014, to Friday, 25 July 2014 were taken as test data. We fitted the spatiotemporal gradient-boosted regression tree (STGBRT) using different numbers of trees (1–5000) and various learning rates (0.01–1) to training data reflecting probe vehicle spatiotemporal characteristics extracted from the urban road network. To evaluate the performance of an STGBRT model that combines various parameters, we introduced the mean absolute percentage error (MAPE) as an indicator. The definition of the MAPE is as follows:

$$MAPE = 100\% \times \frac{1}{n}\sum_{i=1}^{n}\frac{\left|t_{pv,i} - t_{true,i}\right|}{t_{true,i}} \tag{15}$$

where $t_{pv,i}$ denotes the link travel time prediction for a probe vehicle traveling the target link at some future time and $t_{true,i}$ is the true link travel time.

To study the influence of the number of trees and the learning rate on prediction accuracy, we conducted experiments using different numbers of trees. Figures 9 and 10 demonstrate the influence of various parameters including the number of trees (N) and the learning rate (lr) on the link travel time prediction errors using the MAPE. Here, the parameter N represents the number of basic trees in the STGBRT model and lr denotes learning rate. Theoretically, higher prediction accuracy can be achieved by increasing the number of trees in the model. When there are too many trees, however, overfitting may occur. This overfitting influences the prediction accuracy of the model when applied to probe vehicle travel time data that were not included in the training dataset. At the same time, the computing time for the model will increase with the number of basic trees included in the model. Figure 9 plots the relationship between MAPE and N under different learning rates. The lower panel of Figure 9 shows a portion of the upper panel in greater detail. As is shown, MAPE decreases as the number of regression trees increases, up to a certain value. The slopes of the plotted curves vary with different learning rates, lr. The curve for $lr = 0.01$ has the smallest slope because the contribution to prediction accuracy from every tree becomes limited with a small learning rate. It reaches a minimum with N = 300. The curves corresponding to higher learning rates decline more quickly and quickly reach the minimum MAPE using basic trees. For example, the curved with *lr = 0.5* and *lr = 1* reach a minimum at *N = 10* and *N = 50*, respectively. As we can see from Figure 9, higher learning rates such as *lr = 1*, *lr = 0.5*, *lr =0.25*, and *lr = 0.2* obtain the best predicted performance with relatively few regression trees. Too many trees may lead to overfitting if the number of regression trees exceeds some threshold. Consequently, we can guarantee prediction accuracy by using enough trees and at the same time prevent overfitting with an appropriate number of trees.

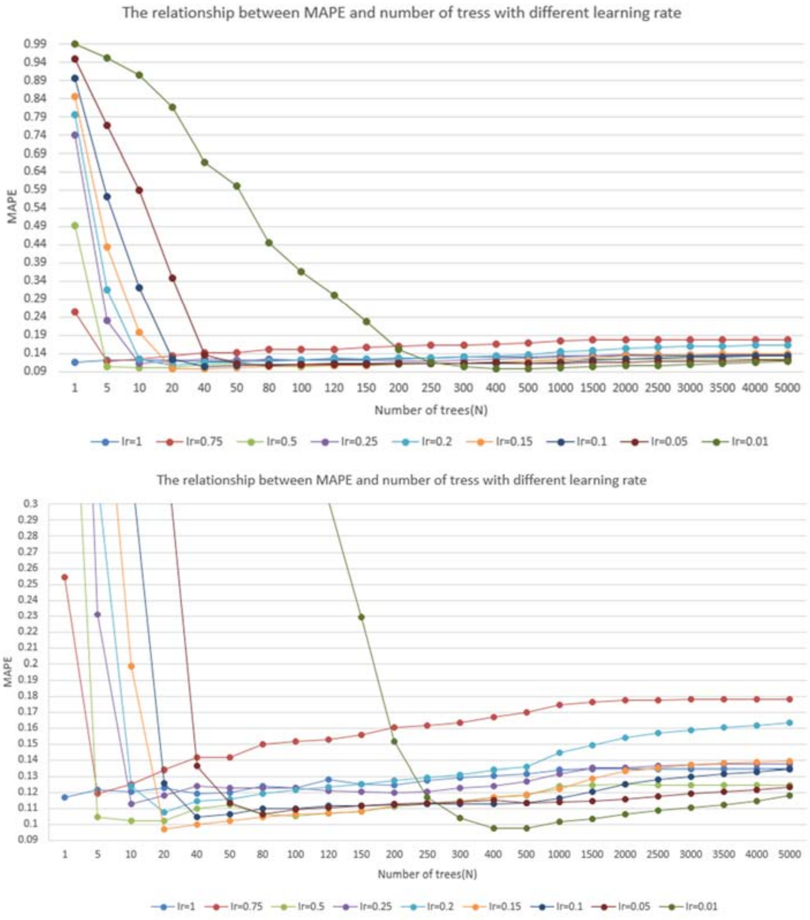


Figure 9. The relationship between MAPE and the number of trees used.

Figure 10 illustrates the effect of learning rate on MAPE. MAPE varies with the learning rate provided that the number of regression trees is held constant. The lower panel of Figure 10 shows a portion of the upper panel in greater detail. The learning rate is used to adjust the influence of each tree on the prediction precision of the model. The learning rate value ranges from 0 to 1. In general, smaller values limit the contribution of each tree to the model accuracy. More iterations are usually required when predicting the link travel time with smaller learning rates. The optimal value of *lr* varies with the number of trees in the ensemble. The MAPE for predicted travel time goes down with an increase in the learning rate if the number of regression trees is 200 or less. In this case, MAPE decreases with an increase in the number of regression trees at the same learning rate. MAPE reached a minimum when the learning rate equaled 0.01 and the number of regression trees exceeded 200. Taking *N = 500* in Figure 10 as an example, MAPE reaches a minimum when *lr = 0.01*, whereas the error increases with the learning rate. This result occurs because the number of regression trees is sufficient; the model reaches its highest accuracy at a smaller learning rate of 0.01; higher learning rates led to poor predictive performance under these conditions.

Table 8. Sample rows from the training and testing datasets (that is, inputs and output for the models).

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**
Weekday	Period of day	tarHTT$_{t-1}$	tarHTT$_{t-2}$	ΔtarHTT $_{t-1}$	tarRTT$_{t-1}$	tarRTT$_{t-2}$	ΔtarRTT $_{t-1}$	UpHTT$_{t-1}$	UpHTT$_{t-2}$	ΔUpHTT $_{t-1}$	UpRTT$_{t-1}$	UpRTT$_{t-2}$	ΔUpRTT $_{t-1}$	DoHTT$_{t-1}$	DoHTT$_{t-2}$	ΔDoHTT $_{t-1}$	tarRTT$_{t}$
1	13.0	111.16	114.01	−2.85	49.0	110.0	−61.0	85.56	85.56	0.0	85.56	85.56	0.0	86.26	86.26	0.0	105.82
2	20.0	143.01	209.02	−66.01	153.83	111.14	42.69	96.49	110.89	−14.4	71.0	110.89	−39.89	103.91	114.3	−10.39	239.41
3	30.0	109.18	113.22	−4.04	102.97	175.2	−72.23	75.87	78.24	−2.37	160.0	58.0	102.0	106.32	89.06	17.26	144.19
4	36.0	132.91	125.41	7.5	286.39	237.0	49.39	74.68	89.75	−15.07	162.99	89.75	73.24	121.74	109.14	12.6	88.0
5	15.0	98.81	98.81	0.0	47.97	106.0	−58.03	73.87	70.79	3.08	73.0	70.79	2.21	101.6	85.72	15.88	130.47

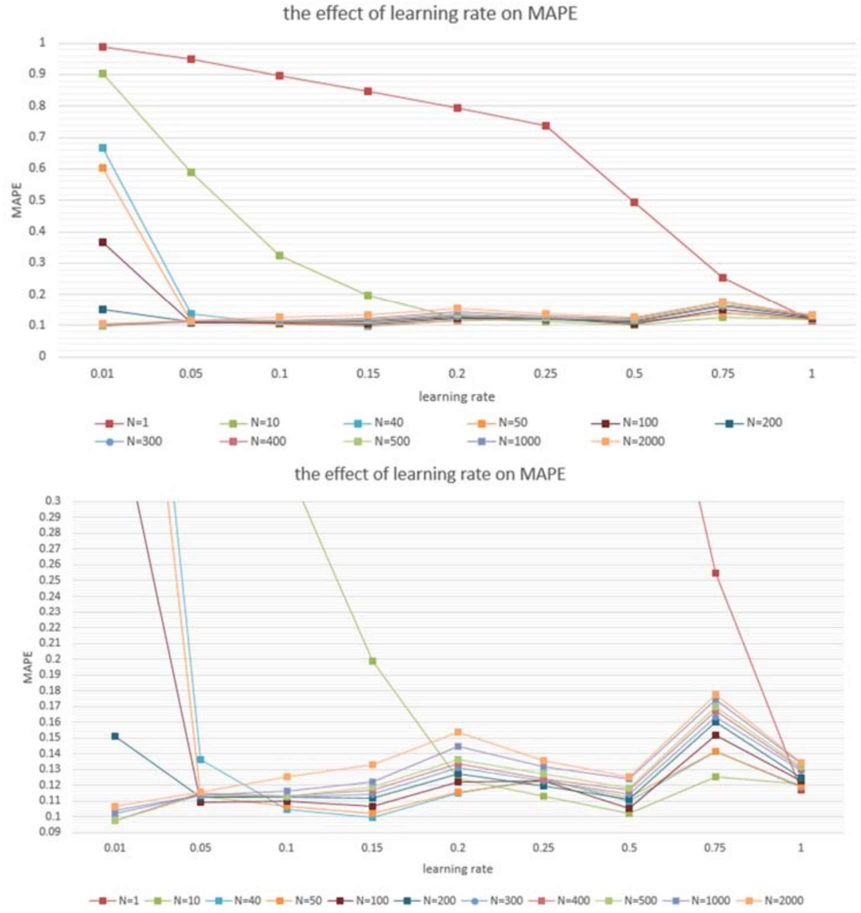


Figure 10. Line chart showing the effect of learning rate on MAPE.

Figure 11 shows a flowchart that describes how the GBRT model predicts link travel time while including information from spatiotemporal correlations. Based on our experimental results, we can draw the following conclusions. (1) A smaller learning rate with more basic regression trees in the model for prediction accuracy is superior to a larger learning rate with fewer basic regression trees. A smaller learning rate shrinks the contribution of each tree to the model prediction accuracy and achieves optimal prediction performance with more reliable prediction results. (2) It is necessary to find a balance between prediction accuracy and computational time. A small learning rate combined with a greater number of basic regression trees will need more computational time to reach the same performance, while lower prediction accuracy requires less computation time. In our experiment, MAPE reached a minimum when the learning rate was 0.01 and the number of regression trees was 500. Consequently, we trained the STGBRT model using those parameters to accurately predict link travel time.

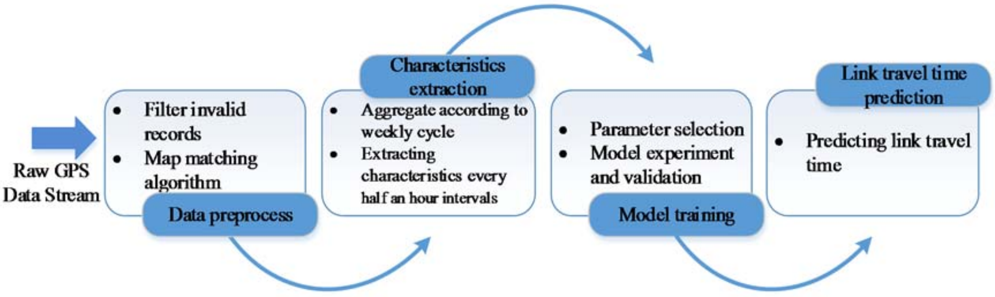

Raw GPS
Data Stream
• Filter invalid records
• Map matching algorithm
Data preprocess
Characteristics extraction
• Aggregate according to weekly cycle
• Extracting characteristics every half an hour intervals
• Parameter selection
• Model experiment and validation
Model training
Link travel time prediction
• Predicting link travel time

Figure 11. Flowchart showing the procedure used by the GBRT model to predict link travel time.

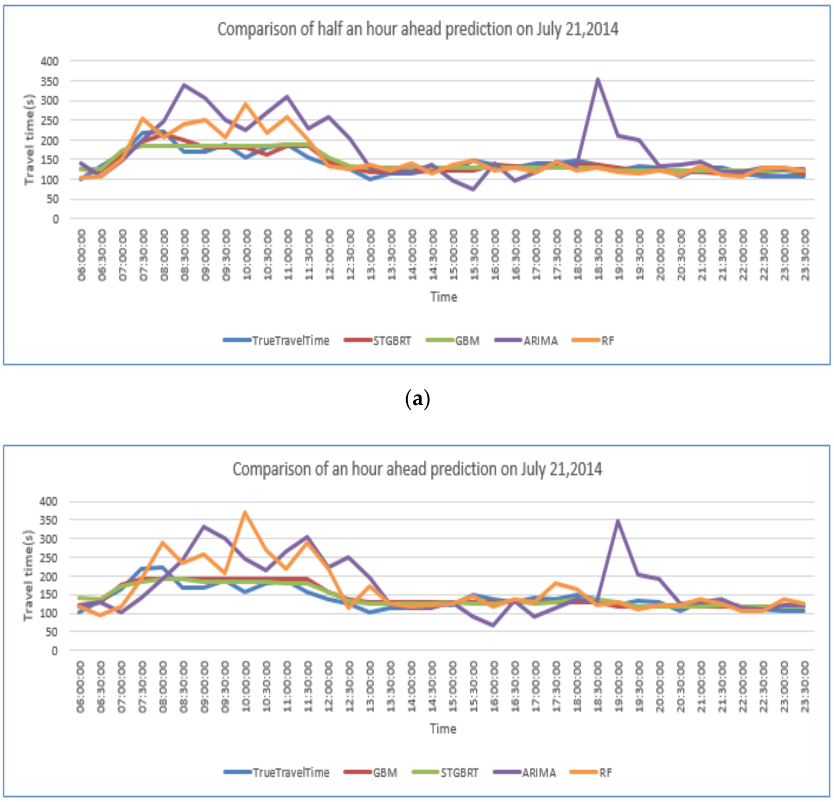
4.3. Model Comparisons

To test the performance of the spatiotemporal gradient–boosted regression tree (STGBRT) method, we compared the predictive performance of STGBRT with that of the Autoregressive Integrated Moving Average [12], Random Forest [54] and Gradient Boosting [27] methods in terms of their absolute percentage errors (MAPE). The Gradient Boosting Method (GBM) considers the time correlation of a target link without regard for the influence of spatial correlation or big data describing historic traffic conditions in estimating link travel time. The Autoregressive Integrated Moving Average Model (ARIMA) model is a generalization of the autoregressive moving average (ARMA) model and is one of the most widely recognized methods for traffic parameter forecasting. The model is fitted to time series data to understand the data better or to predict future points in the series. ARIMA is applied in cases where data show evidence of non-stationarity. It converts non-stationary time series to stationary time series. The model is constructed using the dependent variable, its lag value, and the present value of the random error; predictions from ARIMA are based on regression of current and past data. Non-seasonal ARIMA models are generally denoted as ARIMA (p, d, q) where parameters p, d, and q are non-negative integers, p is the order of the autoregressive model, d is the degree of differencing, and q is the order of the moving average model. Optimization of the ARIMA model involves order selection and parameter estimation. Detailed information on the theoretical background underlying ARIMA, and the steps involved in fitting an ARIMA model, can be found in the literature [55]. The Random Forest (RF) method is another widely used ensemble method whose extension was developed by Leo Breiman [54] and is different from the gradient–boosted regression tree method.

To compare these four methods for predicting link travel time, we obtained statistical data collected by probe vehicles traversing the regional road network in Wuhan on weekdays, Monday to Friday, except holidays, from January to May 2014. We extracted the spatiotemporal features of links within the network. The data from 21 July 2014 to 22 July 2014 were used as test data to compare the prediction performance among the four models (STGBRT, GBM, RF, and ARIMA). The prediction accuracy of these four models was compared based on their predictions one and two time steps (that is, 30 and 60 min) after the present time. The experiment discussed in Section 4.2 showed that the MAPE of the STGBRT model achieved a minimum value when the learning rate was set to 0.01 and the number of basic regression trees was set to 500. Therefore, in the comparative experiment, we set the corresponding experimental parameters to 0.01 and 500, respectively. For GBM and ARIMA, we tested different combinations of variables during the training process and selected the parameters that achieved the minimum MAPE values.

We used traffic big data representing historic traffic conditions from January to May in 2014 and real data obtained from the 11 weeks between 5 May 2014 and 20 July 2014 as training data. We used two days of data (21 and 22 July 2014) as test data to compare the prediction performance among STGBRT, GBM, and ARIMA. The line charts in Figures 12 and 13 illustrate the variation among predictions made 30 min and one hour ahead from the four models on 21 July 2014 and 22 July 2014, respectively. The blue line in the two figures represents the true link travel time, while the red line represents prediction results from the STGBRT model, the green line represents the prediction results from GBM, the orange line represents the prediction from RM and the purple line represents the prediction results from the ARIMA model. As shown, the STGBRT model and GBM model fit the true link travel time most closely. ARIMA provided the least favorable match to the true link travel time among the four models. Under the same conditions, the predictions of STGBRT outperform those from the random forest method in our experiments, as depicted in Figures 12–14. Figure 14 shows a comparison of the MAPE values for the performance of these four models for predictions made 30 min and one hour ahead. As illustrated in Figure 14, the prediction results of STGBRT outperformed those of the other three models. The MAPE for STGBRT (7.43%) was superior to the MAPE values corresponding to half-an-hour predictions for GBM, RF, and ARIMA, which were 9.37%, 15.83%, and 33.79%, respectively. At the same time, the STGBRT half-an-hour prediction performance had a significantly better MAPE value (7.43%) than the one-hour prediction (9.49%). Figure 15 illustrates the standard deviations of predictions made 30 min and one hour ahead by the four models for 21 July

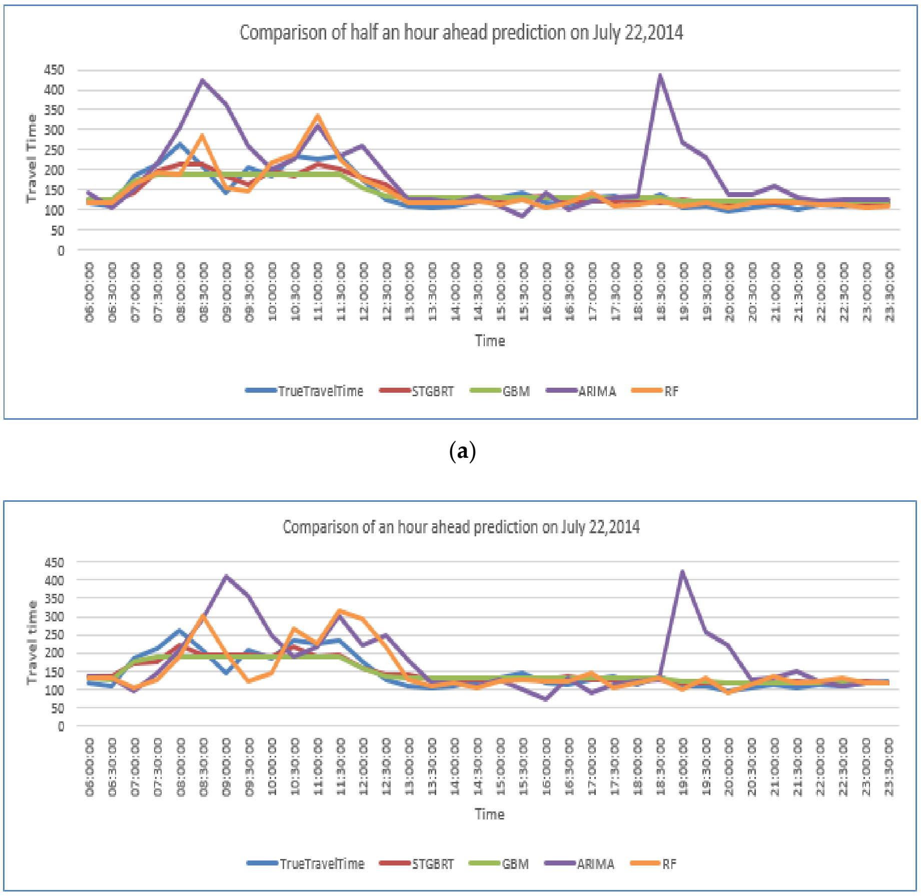
2014 and 22 July 2014. As illustrated in Figure 15, the prediction results of STGBRT had a small MAPE value and outperformed the other three models except in terms of the one-hour predictions made for 21 July 2014. Figure 16 gives the computational performance of different models under the same conditions, that is, using the same training and prediction data. The figure shows that STGBRT, GBM, and RF require similar amounts of computational time: 5.09 s, 5.73 s, and 5.24 s, respectively. The ARIMA model requires the smallest amount of computation time; however, it had poor prediction performance compared to the other three models, as depicted in Figure 14. A Wilcoxon test showed that the differences between true link travel time and the results from the STGBRT, GBM, and RF models are all symmetrically distributed about zero except for predictions made one hour ahead by the RF model for 21 July 2014. However, the differences between true link travel time and predicted values from ARIMA are not symmetrically distributed about zero except for predictions made one hour ahead for 22 July 2014. Therefore, the STGBRT, GBM, and RF models yield better predictions than the ARIMA model. Figure 17 shows five days (Monday, 21 July 2014–Friday, 25 July 2014) of predicted link travel times from the STGBRT model. The blue line represents the true link travel time, and the red line represents the predicted link travel time. Table 9 shows the MAPE values for the travel time prediction obtained from the STGBRT model from Monday to Friday; the STGBRT model had high MAPE values. Figure 17 reflects overall trends, as well as how well the models captured sudden changes in travel time. For example, on 21 July 2014 (upper panel of Figure 17), the STGBRT model captured changes especially well during the morning rush hour when congestion is likely to occur. Theoretically, the STGBRT model can handle complex interactions among input variables and can fit the complex nonlinear relationships found in dynamic traffic systems for superior prediction performance.



(**a**)

(**b**)

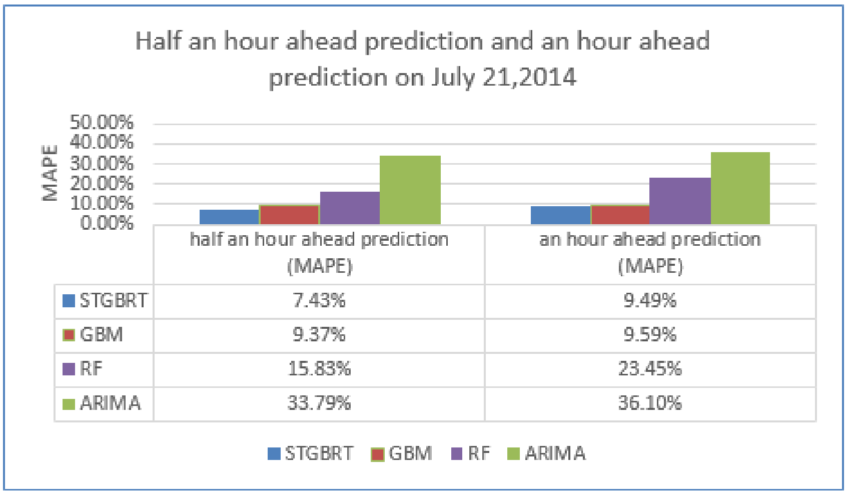
Figure 12. Comparisons of predictions made 30 min and one hour ahead for the four models, using data from 21 July 2014. (**a**) Comparisons of predictions made 30 min ahead; (**b**) Comparisons of predictions made one hour ahead.



(**a**)

(**b**)

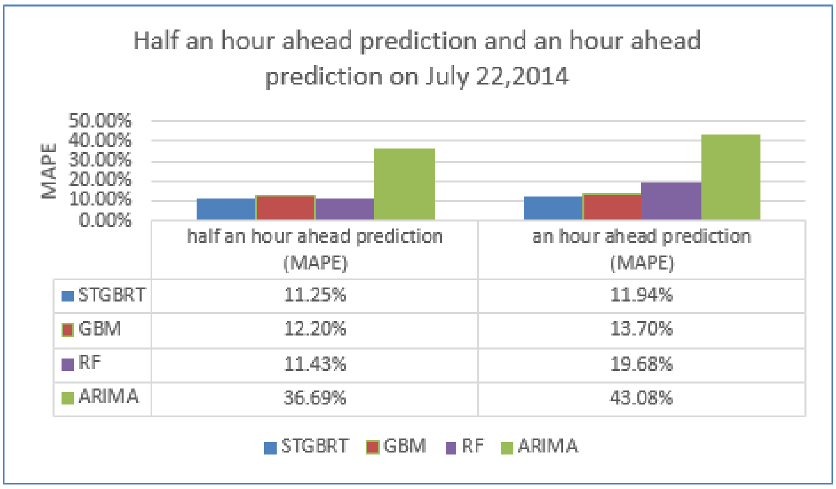
Figure 13. Comparisons of predictions made 30 min and one hour ahead for the four models, using data from 22 July 2014. (**a**) Comparisons of predictions made 30 min ahead; (**b**) Comparisons of predictions made one hour ahead.



	half an hour ahead prediction (MAPE)	an hour ahead prediction (MAPE)
STGBRT	7.43%	9.49%
GBM	9.37%	9.59%
RF	15.83%	23.45%
ARIMA	33.79%	36.10%

(**a**)

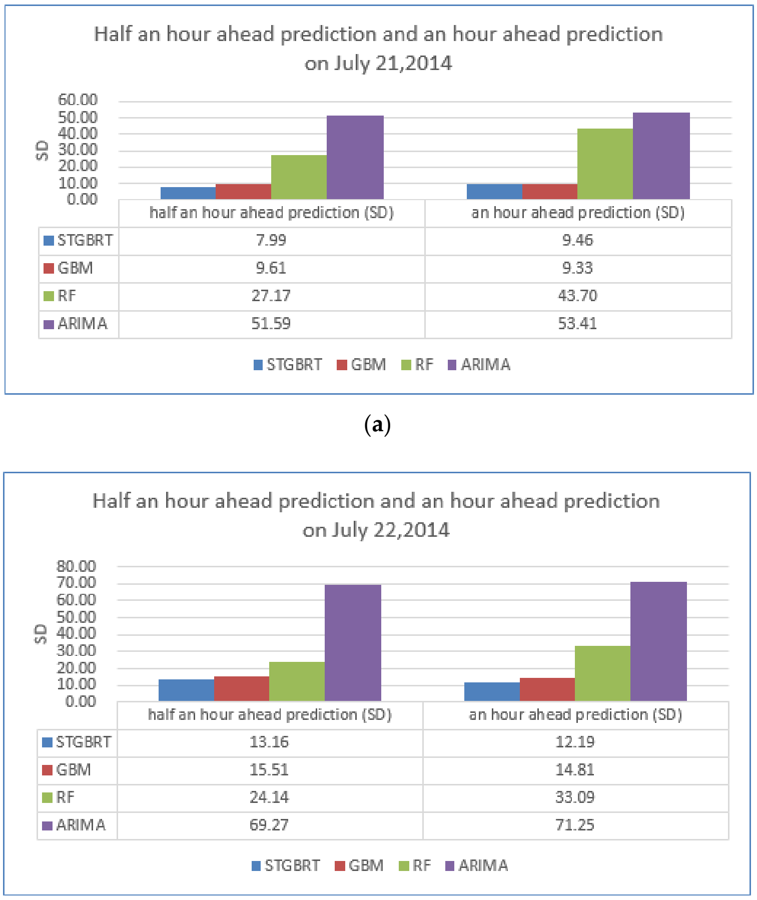
Figure 14. *Cont.*



	half an hour ahead prediction (MAPE)	an hour ahead prediction (MAPE)
STGBRT	11.25%	11.94%
GBM	12.20%	13.70%
RF	11.43%	19.68%
ARIMA	36.69%	43.08%

(**b**)

Figure 14. Comparison of MAPEs for predictions made 30 min and one hour ahead, generated by STGBRT, GBM, and ARIMA. (**a**) Comparison of MAPEs for predictions made 30 min ahead; (**b**) Comparison of MAPEs for predictions made one hour ahead.



	half an hour ahead prediction (SD)	an hour ahead prediction (SD)
STGBRT	7.99	9.46
GBM	9.61	9.33
RF	27.17	43.70
ARIMA	51.59	53.41

(**a**)

	half an hour ahead prediction (SD)	an hour ahead prediction (SD)
STGBRT	13.16	12.19
GBM	15.51	14.81
RF	24.14	33.09
ARIMA	69.27	71.25

(**b**)

Figure 15. Comparison of standard deviations of predictions made 30 min and one hour ahead, generated by STGBRT, GBM, and ARIMA. (**a**) Comparison of standard deviations of predictions made 30 min ahead; (**b**) Comparison of standard deviations of predictions made one hour ahead.

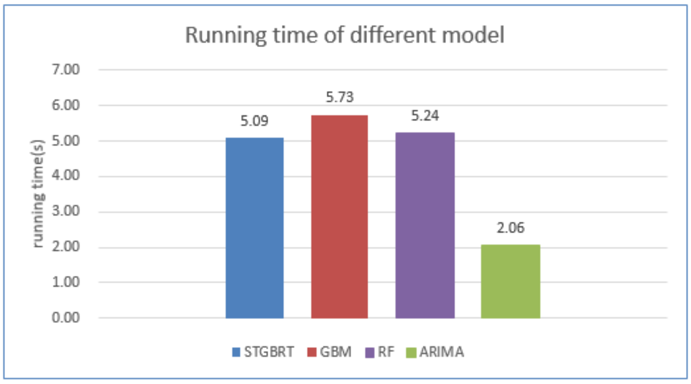
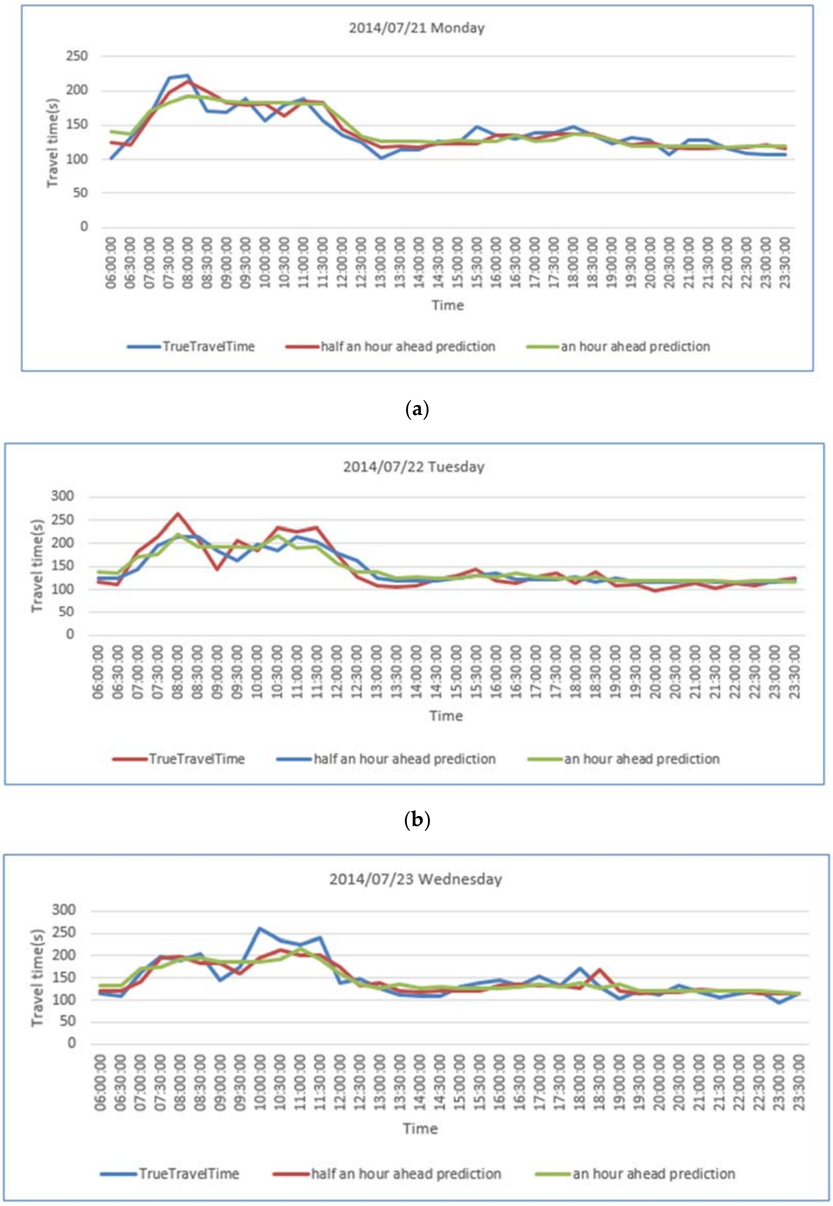


Figure 16. Computational time required by STGBRT, GBM, RF, and ARIMA.

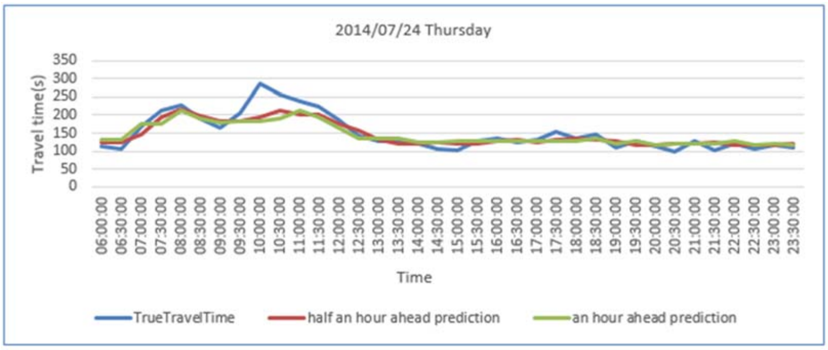


(**a**)

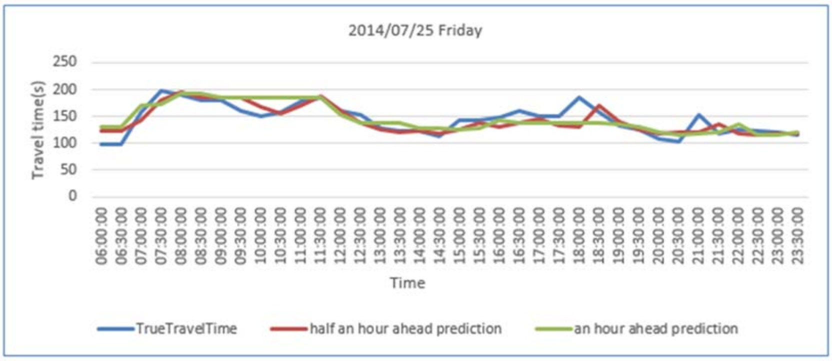
(**b**)

(c)

Figure 17. *Cont.*



(d)



(e)

Figure 17. Travel time prediction results from the STGBRT model from Monday to Friday. (**a**) Monday; (**b**) Tuesday; (**c**) Wednesday; (**d**) Thursday; (**e**) Friday.

Table 9. The MAPE of travel time prediction results from the STGBRT model from Monday to Friday.

Ate	21 July 2014	22 July 2014	23 July 2014	24 July 2014	25 July 2014
MAPE of predictions made 30 min ahead	7.43%	11.25%	11.23%	10.26%	7.89%
MAPE of predictions made an hour ahead	9.49%	11.94%	10.98%	10.31%	9.77%

5. Discussion and Conclusions

The GBRT model has characteristics that make it different from traditional ensemble methods, such as the random forest and bagged trees approaches, as well as classical statistical approaches. The GBRT model grows trees sequentially by adjusting the weight of the training data distribution in the direction of "steepest descent" to minimize the loss function. It reduces model bias through forward stepwise modeling and reduces variance through averaging. However, our proposed method, the STGBRT-based travel time prediction model, has considerable advantages over the traditional GBRT model. The proposed method not only uses the "steepest descent" method but also incorporates the spatiotemporal correlation between a target link and adjacent links in the training data. Thus, it delivers higher performance than the GBM, ARIMA, or RF models in terms of prediction accuracy.

As far as the authors are aware, there are few studies that discuss the STGBRT method in the context of travel time prediction and little work on the application of the STGBRT method to estimate urban link travel time. The STGBRT model can capture sudden discontinuities, an important characteristic of traffic flows, given that traffic changes quickly from uncongested to congested and vice versa. More importantly, the STGBRT model considers the spatiotemporal features of traffic, not only present traffic flows but also in relation to historical traffic data. It not only considers target link traffic features but also exploits information from adjacent traffic link features. In contrast to traditional machine learning algorithms, which are often regarded as "black boxes", the number of basic regression trees and the learning rate in STGBRT are parameters that can be analyzed and set. Compared with the GBM and ARIMA methods, the STGBRT method considers spatiotemporal features and is superior to conventional statistical models.

Parameter optimization is an important aspect for link travel time prediction using the STGBRT model. Just as in model optimization, the performance of the STGBRT model is substantially influenced by its parameters, including the number of regression trees, the learning rate, and the complexity of the tree. Therefore, it is necessary to find the optimal combination of variables when using the STGBRT model. Computation time is another important issue when increasing the number and the complexity of the regression trees. Consequently, we must weigh the increase in calculation time against the accuracy of model.

The STGBRT model has distinct advantages in terms of free-flow travel time prediction. It is possible for us to collect large quantities of different traffic data from road sensors, smart phones, and GPS devices, given the development of these advanced technologies. As time goes on, more and more traffic information can be collected and used to study traffic phenomena. Therefore, it is critical to find a model that can represent complex relationships when combining heterogeneous big data. The STGBRT model can address complex nonlinear relationships, making it a promising algorithm for travel time prediction. The accuracy of the proposed modeling approach is such that it can be applied in intelligent transportation systems for link travel time prediction or real-time travel time prediction. It can also be extended to traffic flow prediction. This model, however, currently only considers first-order spatial correlations of target links. Further research will incorporate second-order and higher levels of correlation to more accurately capture traffic dynamics. Another issue that must be addressed is the lack of data. When historical and real-time traffic data for the same time are missing, this model cannot predict link travel. This problem is an important topic that we will investigate in the future. Our experimental results are based on specific road segments. We will extend our experiments to other road segments in the future.

Acknowledgments: This work was supported by the National Natural Science Foundation of China (Grant No. 41271401) and the National 863 projects "Multi-source Information Real-time Access and Heterogeneous Information Autonomous Loading Technology under the Unified Spatiotemporal System" (Grant No. 2013AA122301) and "Pan-Information Map Fusion Technology Based on the Internet Superposition Protocol" (Grant No. 2013AA12A203). Additional support was provided by the National Science and Technology Support Plan through the project "Key Technology and Applications of Location-based Sensor Network and Pan-information Map" (Grant No. 2012BAH35B03).

Author Contributions: Faming Zhang performed the research, analyzed the data and wrote the paper. Xinyan Zhu co-designed the research. Tao Hu co-designed the research and extensively updated the paper. Wei Guo developed some earlier prototypes. Chen Chen and Lingjia Liu edited the paper. All authors read and approved the final manuscript.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Liu, K.; Yamamoto, T.; Morikawa, T. Feasibility of using taxi dispatch system as probes for collecting traffic information. *J. Intell. Transp. Syst. Technol. Plan. Oper.* **2009**, *13*, 16–27. [CrossRef]
2. Wang, Y.; Zheng, Y.; Xue, Y. Travel time estimation of a path using sparse trajectories. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014.
3. Li, J. Estimation and Prediction of Link Travel Time for Urban Trunk and Secondary Street. Ph.D. Thesis, Jilin University, Jilin, China, 2012. (in Chinese)
4. Yao, E.J.; Zuo, T. Real-time map matching algorithm based on low-sampling-rate probe vehicle data. *Beijing J. Beijing Univ. Technol.* **2012**, *39*, 909–913.
5. Zheng, Y.; Liu, Y.; Yuan, J.; Xie, X. Urban computing with taxicabs. In Proceedings of the 13th International Conference on Ubiquitous Computing, Beijing, China, 17–21 September 2011.

6. Zheng, Y.; Liu, F.; Hsieh, H.P. U-Air: When urban air quality inference meets big data. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Sydney, Australia, 10–13 August 2013.
7. Min, W.; Wynter, L. Real-time road traffic prediction with spatio-temporal correlations. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 606–616. [CrossRef]
8. Fei, X.; Lu, C.C.; Liu, K. A Bayesian dynamic linear model approach for real-time short-term freeway travel time prediction. *Transp. Res. Part C Emerg. Technol.* **2011**, *19*, 1306–1318. [CrossRef]
9. Li, L.; Li, Y.; Li, Z. Efficient missing data imputing for traffic flow by considering temporal and spatial dependence. *Transp. Res. Part C Emerg. Technol.* **2013**, *34*, 108–120. [CrossRef]
10. Haghani, A.; Hamedi, M.; Sadabadi, K.F. Freeway travel time ground truth data collection using bluetooth sensors. *J. Transp. Res. Board* **2010**, *2160*, 60–68. [CrossRef]
11. Williams, B.; Durvasula, P.; Brown, D. Urban freeway traffic flow prediction: Application of seasonal autoregressive integrated moving average and exponential smoothing models. *Transp. Res. Rec. J. Transp. Res. Board* **1998**, *1644*, 132–141. [CrossRef]
12. Smith, B.; Demetsky, M. Traffic flow forecasting: Comparison of modeling approaches. *J. Transp. Eng.* **1997**, *123*, 261–266. [CrossRef]
13. Wang, J.; Shi, Q. Short-term traffic speed forecasting hybrid model based on Chaos-wavelet analysis-support vector machine theory. *Transp. Res. Part C Emerg. Technol.* **2012**, *27*, 219–232. [CrossRef]
14. Wei, Y.; Chen, M.C. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transp. Res. Part C Emerg. Technol.* **2012**, *21*, 148–162. [CrossRef]
15. Li, L.; Chen, X.; Li, Z.; Zhang, L. Freeway travel-time estimation based on temporal–spatial queueing model. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 1536–1541. [CrossRef]
16. Zhang, Y.; Zhang, Y.; Haghani, A. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 65–78. [CrossRef]
17. Montgomery, D.C.; Jennings, C.L.; Kulahci, M. *Introduction to Time Series Analysis and Forecasting*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
18. Hong, W.C. Traffic flow forecasting by seasonal SVR with chaotic simulated annealing algorithm. *Neurocomputing* **2011**, *74*, 2096–2107. [CrossRef]
19. Van Hinsbergen, C.; Van Lint, J.; Van Zuylen, H. Bayesian committee of neural networks to predict travel times with confidence intervals. *Transp. Res. Part C Emerg. Technol.* **2009**, *17*, 498–509. [CrossRef]
20. Antoniou, C.; Koutsopoulos, H.N.; Yannis, G. Dynamic data-driven local traffic state estimation and prediction. *Transp. Res. Part C Emerg. Technol.* **2013**, *34*, 89–107. [CrossRef]
21. Vlahogianni, E.I.; Karlaftis, M.G.; Golias, J.C. Short-term traffic forecasting: Where we are and where we're going. *Transp. Res. Part C Emerg. Technol.* **2014**, *43*, 3–19. [CrossRef]
22. Zhou, Z.H. *Ensemble Methods: Foundations and Algorithms*; CRC Press: Boca Raton, FL, USA, 2012.
23. Hamner, B. Predicting travel times with context-dependent random forests by modeling local and aggregate traffic flow. In Proceedings of the ICDMW 2010, Sydney, Australia, 14–17 December 2010.
24. Wang, Y. Prediction of weather impacted airport capacity using ensemble learning. In Proceedings of the Digital Avionics Systems Conference (DASC) 2011, Sacramento, CA, USA, 25–29 September 2011.
25. Ahmed, M.M.; Abdel-Aty, M. Application of stochastic gradient boosting technique to enhance reliability of real-time risk assessment. *Transp. Res. Rec. J. Transp. Res. Board* **2013**, *2386*, 26–34. [CrossRef]
26. Chung, Y.S. Factor complexity of crash occurrence: An empirical demonstration using boosted regression trees. *Accid. Anal. Prev.* **2013**, *61*, 107–118. [CrossRef] [PubMed]
27. Zhang, Y.; Haghani, A. A gradient boosting method to improve travel time prediction. *Transp. Res. Part C Emerg. Technol.* **2015**. [CrossRef]
28. Polikar, R. Ensemble based systems in decision making. *IEEE Circ. Syst. Mag.* **2006**, *6*, 21–45. [CrossRef]
29. Leistner, C.; Saffari, A.; Santner, J.; Bischof, H. Semi-supervised random forests. In Proceedings of the IEEE 12th International Conference on Computer Vision, Porto, Portugal, 27 February–1 March 2009.
30. Strobl, C.; Malley, J.; Tutz, G. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychol. Method.* **2009**, *14*, 323–348. [CrossRef] [PubMed]
31. Hastie, T.; Tibshirani, R.; Friedman, J. Unsupervised learning. In *The Elements of Statistical Learning*; Springer: Berlin/Hamburg, Germany, 2009; pp. 485–585.

32. Quinlan, J.R. Induction of decision trees. *Mach. Learn.* **1986**, *1*, 81–106. [CrossRef]
33. Ruiz, R.; Stützle, T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *Eur. J. Operat. Res.* **2007**, *177*, 2033–2049. [CrossRef]
34. Breiman, L. Arcing the Edge. Technical Report 486; Statistics Department, University of California at Berkeley: Berkeley, CA, USA, 1997.
35. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Annal. Stat.* **2001**, 1189–1232. [CrossRef]
36. Friedman, J.H. Stochastic gradient boosting. *Comput. Stat. Data Anal.* **2002**, *38*, 367–378. [CrossRef]
37. Mason, L.; Baxter, J.; Bartlett, P.L.; Frean, M. Boosting algorithms as gradient descent in function space. In Proceedings of the NIPS 1999, Denver, CO, USA, 29 November–4 December 1999.
38. Natekin, A.; Knoll, A. Gradient boosting machines, a tutorial. *Front. Neurorobot.* **2013**. [CrossRef] [PubMed]
39. Soper, H.E.; Young, A.W.; Cave, B.M.; Lee, A.; Pearson, K. On the distribution of the correlation coefficient in small samples. Appendix II to the papers of" Student" and RA Fisher. *Biometrika* **1917**, *11*, 328–413.
40. Box, G.; Jenkins, G. *Time Series Analysis: Forecasting and Control*; Holden-Day: San Francisco, CA, USA, 1970.
41. Mori, U.; Mendiburu, A.; Álvarez, M.; Lozano, J.A. A review of travel time estimation and forecasting for advanced traveller information systems. *Transp. A Transp. Sci.* **2015**, *11*, 119–157. [CrossRef]
42. Fouque, C.; Bonnifait, P. Matching raw GPS measurements on a navigable map without computing a global position. *IEEE Trans. Intell. Transp. Syst.* **2012**, *13*, 887–898. [CrossRef]
43. Chen, B.Y.; Yuan, H.; Li, Q.; Lam, W.H.; Shaw, S.L.; Yan, K. Map-matching algorithm for large-scale low-frequency floating car data. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 22–38. [CrossRef]
44. Yuan, J.; Zheng, Y.; Zhang, C.; Xie, X.; Sun, G.Z. An interactive-voting based map matching algorithm. In Proceedings of the Eleventh International Conference on Mobile Data Management, Kansas City, MI, USA, 23–26 May 2010.
45. Zhang, Y.; Yang, B.; Luan, X. Automated matching urban road networks using probabilistic relaxation. *Acta Geod. Catogr. Sin.* **2012**, *41*, 933–939.
46. Li, Q.; Hu, B.; Yue, Y. Flowing car data map-matching based on constrained shortest path algorithm. *Geomat. Inf. Sci. Wuhan Univ.* **2013**, *7*, 805–808.
47. Yu, D.X.; Gao, X.Y.; Yang, Z.S. Individual vehicle travel-time estimation based on GPS data and analysis of vehicle running characteristics. *J. Jilin Univ.* **2010**, *40*, 965–970. (in Chinese)
48. Dong, H.; Wu, F. Estimation of average link travel time using fuzzy C-mean. *Bull. Sci. Technol.* **2011**, *27*, 426–430.
49. Jiang, G.; Chang, A.; Zhang, W. Comparison of link travel-time estimation methods based on GPS equipped floating car. *J. Jilin Univ.* **2009**, *39*, 182–186. (in Chinese)
50. Liu, Y.; Kang, C.; Gao, S.; Xiao, Y.; Tian, Y. Understanding intra-urban trip patterns from taxi trajectory data. *J. Geogr. Syst.* **2012**, *14*, 463–483. [CrossRef]
51. Liu, X.; Gong, L.; Gong, Y.; Liu, Y. Revealing travel patterns and city structure with taxi trip data. *J. Transp. Geogr.* **2013**, *43*, 78–90. [CrossRef]
52. Zhang, F.; Zhu, X.; Guo, W.; Ye, X.; Hu, T.; Huang, L. Analyzing urban human mobility patterns through a thematic model at a finer scale. *ISPRS Int. J. Geo-Inf.* **2016**. [CrossRef]
53. Fang, Z.; Li, Q.; Shaw, S.L. What about people in pedestrian navigation? *Geo-spat. Inf. Sci.* **2015**, *18*, 135–150. [CrossRef]
54. Breiman, L. Random forests. *Machine Learn.* **2001**, *45*, 5–32. [CrossRef]
55. Tsay, R.S. *Analysis of Financial Time Series*; John Wiley & Sons: New York, NY, USA, 2005.