sustainability

MDPI

Article

Estimating Toll Road Travel Times Using Segment-Based Data Imputation

Krit Jedwanna [1], Chuthathip Athan [2] and Saroch Boonsiripant [3,*]

[1] Department of Civil Engineering, Faculty of Engineering Rajamangala, University of Technology Phra Nakhon, Bangkok 10300, Thailand; krit.j@rmutp.ac.th
[2] Mobinary Company Limited, Bangkok 10400, Thailand; chuthathip@mobinary.org
[3] Department of Civil Engineering, Faculty of Engineering, Kasetsart University, Bangkok 10900, Thailand
* Correspondence: saroch.b@ku.th; Tel.: +66-81-885-9300

Abstract: Efficient and sustainable transportation is crucial for addressing the environmental and social challenges associated with urban mobility. Accurate estimation of travel time plays a pivotal role in traffic management and trip planning. This study focused on leveraging machine learning models to enhance travel time estimation accuracy on toll roads under diverse traffic conditions. Two models were developed for travel time estimation under a variety of traffic conditions on the Don Muang Tollway, Bangkok, Thailand: a long short-term memory (LSTM) recurrent neural network model and a support vector regression (SVR) model. Missing data were treated using the proposed segment-based data imputation method. Unlike other studies, the effects of missing input data on the travel time model performance were also analyzed. Traffic parameters, such as speed and flow, along with other relevant parameters (time of day, day of the week, holiday indicators, and a missing data indicator), were fed into each model to estimate travel time on each of the four specific routes. The LSTM and SVR results had similar performance levels based on evaluating the all-day pooled data. However, the mean absolute percentage errors were lower for LSTM during peak periods, while SVR performed slightly better during off-peak periods. Additionally, LSTM coped substantially better than SVR with unusual traffic fluctuations. The sensitivity analysis of the missing input data in this study also revealed that the LSTM model was more robust to the high degree of missing data than the SVR model.

Keywords: travel time estimation; long short-term memory applications; support vector regression applications; toll roads

check for updates

Citation: Jedwanna, K.; Athan, C.; Boonsiripant, S. Estimating Toll Road Travel Times Using Segment-Based Data Imputation. *Sustainability* **2023**, *15*, 13042. https://doi.org/10.3390/su151713042

Academic Editor: Lei Miao

Received: 5 April 2023
Revised: 15 August 2023
Accepted: 20 August 2023
Published: 29 August 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Travel time, a key parameter in intelligent transportation systems, plays a vital role in promoting sustainability. Road operators rely on real-time travel time information to effectively manage traffic in various situations, including normal conditions, roadworks, and incidents. Historical travel time data supports planning processes and facilitates performance evaluation. Additionally, travelers utilize en route travel times within advanced traveler information systems to make informed decisions regarding their journeys, such as choosing the optimal departure time and route. By optimizing travel time information, sustainable transportation practices are fostered, leading to improved traffic management and enhanced travel planning for a greener and more efficient transportation system.

Travel time can be estimated from two approaches: direct measurement for ground truth data and indirect measurement for input data. In the former approach, travel time can be directly determined from the difference in the timestamps at the beginning and ending points on the selected corridor. Any of several systems can be used to match instances of an individual vehicle passing two points of interest: in-vehicle global positioning systems [1,2], automatic license plate recognition, automatic vehicle identification

systems [3], and Bluetooth scanners [4–6]. The accuracy of this approach is affected by the penetration rate [7], defined as the ratio between the number of observed vehicles and the total number of vehicles passing the two points of interest.

In contrast, indirect measurement involves travel time estimation based on spot measurements of traffic parameters along the corridor of interest. Traffic data, such as speed, flow, density, and vehicle occupancy, can be measured from loop detectors [8,9], microwave radars [10], or video image processing systems (VIPSs) [11]. Model development is required to estimate the travel time using the traffic parameters obtained from such sensors. Several modeling techniques have been developed to determine the estimated travel time, such as speed-based estimation models, regression models, and recent machine learning models. In addition, machine learning and deep learning are applied to more complicated data for travel time estimation. The deep learning model performs better since it has more functions and a more complicated architecture than the conventional model.

The main contributions of this paper are (1) comparing the two most popular machine learning techniques for a travel time estimation application, namely, support vector regression (SVR) and long short-term memory (LSTM) models, across a variety of traffic conditions, such as peak and off-peak periods, as well as holiday traffic and (2) evaluating how the missing data affect the travel time model performance.

The rest of this paper is organized into the following sections. Section 2 describes the related studies on the data imputation techniques, various travel time estimation models, and the effect of missing data. Section 3 details the data collection process—Section 4 details data preprocessing. Section 5 describes the model design for travel time estimation. Section 6 presents an experiment that assesses the model's performance and describes its results. In addition, the performance of the models is examined under various missing rates of input data in the same section. Finally, Section 7 presents the discussion and concluding remarks regarding this study.

2. Related Works

With the advancement of sensor technology, various traffic sensor devices can be used to collect large amounts of traffic data for traffic planning and management. Travel time and other traffic data may also have great potential for long-term applications. However, in the real world, hardware, software, and communication problems with traffic surveillance devices may cause large amounts of missing or suspicious data. These discrepancies raise concerns about the reliability of such data. Despite the advancements in modeling techniques, these models still suffer from the lack of complete datasets during the actual implementation. For example, Tan et al. [12] identified more than 5 percent missing data from a performance measurement system (PeMS) traffic flow database. Qu et al. [13] found that the missing data ratio in Beijing was usually around 10 percent, but sometimes it amounted to between 20 and 25 percent for various reasons. To maximize the traffic sensor investment costs, as well as the continuity of data input, data cleaning, and imputation are necessary before developing the travel time estimation and prediction models.

Transportation data is unique due to catastrophic changes in traffic state, sudden speed drop, and correlation with neighbors in time and space intervals. Because of these explicit characteristics of transportation data, the missing data imputation technique must be carefully chosen and needs to provide reliable and accurate data for intelligent transportation systems (ITS), such as travel time estimation. To address this problem, several missing value correction approaches have been proposed [14–17], including historical imputation, temporal, and spatial methods.

First, historical imputation methods [5,14,15] replace missing data points with the mean value of previous values collected at the same position at the same time of day and day of the week. Second, temporal interpolation imputes a missing value by replacing it with the average of its previous and following values at the same site. This algorithm is somewhat accurate when the traffic condition is relatively stable [5]. However, this approach does not apply to a real-time application since the following values are not available at the

current moment in time. Lastly, a hybrid algorithm [5] improves historical imputation methods by using time series analysis to reflect the temporal variation of the traffic data over different times of day and days of the week. In addition, spatial interpolation (known as nearest neighbor imputation) [4,13,14] imputes missing traffic volume at a detector station by replacing it with the average between the traffic data measured at its upstream and downstream stations in the same period. There may be a number of detector stations along a stretch of highway. Whether to use the readings from the same period or not depends upon the distances between stations. However, such methods cannot be applied when there are no data from neighboring roads. Additionally, this assumption may be violated when the sensors are not in the same segment. To overcome this limitation, we adopted the segmental spatial data imputation method from [5,18]. We define the spatial scope to impute missing data based on a group of links with a shared traffic property. A group of links may share a traffic property of similar vehicle populations and be contained in that group because this group would be severed from other link groups via the geometry of the road, such as on/off ramps, from the traffic engineer's perspective. Because a group of links will spatially share a traffic property, missing data imputation can be done simultaneously.

Travel time modeling techniques fall into three groups: speed-based estimation models, regression models, and models based on artificial intelligence technology [19]. Speed-based models [20–22] apply a series of spot measurements of speed from point detectors along the corridor to estimate link travel time. Li et al. [21] developed four models based on spot speed data: an instantaneous model, a time slice model, a dynamic time slice model, and a linear model. All models had similar performance, mostly underestimating travel times. Xiao et al. [22] compared instantaneous model estimates with Bluetooth-based travel times. They found that the mean absolute percentage error (MAPE) of estimation during the peak period was 10 to 20 percent higher than during other periods.

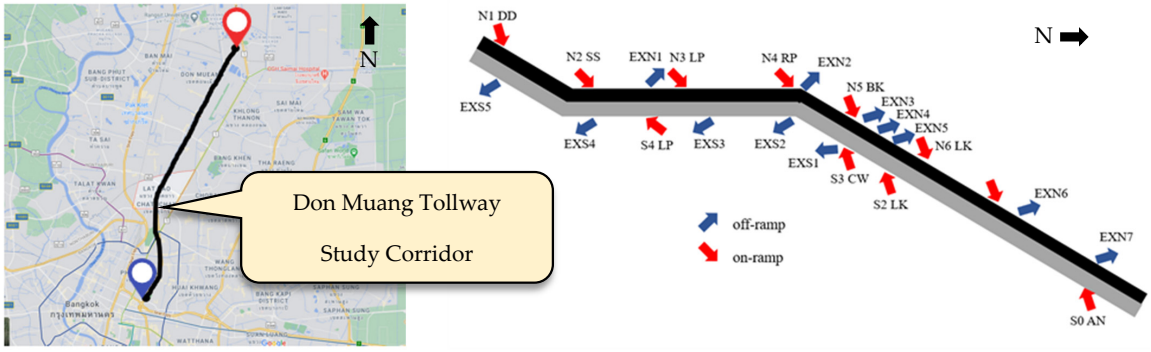
The second category of travel time models is regression models [23,24]. Traffic parameters, such as speed, volume, density, and occupancy, are used as predictor variables to estimate travel time. Historical data can be used in multiple linear regression fitting of the aforementioned traffic parameters to direct measurements of travel time and ground truth.

The third category of travel time models is the set of models based on machine learning methods, in which a computer uses system data to learn the system's behavior without being explicitly programmed [25,26]. There are several machine learning models for the estimation or prediction fields, such as k-nearest neighbor [27], Random Forests [28], and XGBoost [29]. However, these models do not focus on time-series data. The two most common machine learning approaches for time-series data are deep learning and support vector regression (SVR) [30–34]. Hou and Edara [31] deployed a recurrent neural network (RNN) with long short-term memory (LSTM) architecture to predict travel times for a road network in Saint Louis, Missouri, USA. The prediction results showed that deep learning could accurately predict both congested and uncongested traffic conditions. Liu et al. [33] constructed a deep learning model for predicting travel time on freeway sections in California, USA. Their results showed that LSTM was superior to other statistical models, such as linear regression, ridge and lasso regression, autoregressive integrated moving average, and deep neural networks. Going beyond neural network models, Vanajakshi and Rilett [30] constructed a support vector regression model to predict short-time travel times. Their results showed that SVR models outperformed multilayer perceptron artificial neural network models when the disparities between training and testing datasets were high.

In summary, in recent years, SVR and LSTM are two machine learning models that have gained popularity for use with time-series data. However, few studies have investigated their performance across a variety of traffic conditions. In this study, SVR and LSTM models were compared across a variety of traffic conditions, such as peak and off-peak periods, as well as holiday traffic. The objective of this study was to determine the performance of machine learning models in estimating travel time on toll roads under a variety of traffic conditions.

3. Study Corridor

Don Muang Tollway is a 21.9 km toll road connecting central and northern Bangkok, Thailand, as shown in Figure 1. The 6-lane corridor carries approximately 78,000 vehicles per day during weekdays and 46,000 vehicles daily during weekends. The study corridor mainly serves commuting traffic between northern Bangkok and the central city zone. During peak periods, this results in traffic imbalance between the inbound and outbound directions, with the inbound direction having traffic congestion in the morning and a relatively low volume for the rest of the day, while the outbound direction has its peak congestion in the evening and a relatively low volume for the rest of the day. In total, 151 VIPSs and 11 Bluetooth readers were temporarily installed along this corridor to monitor traffic conditions.

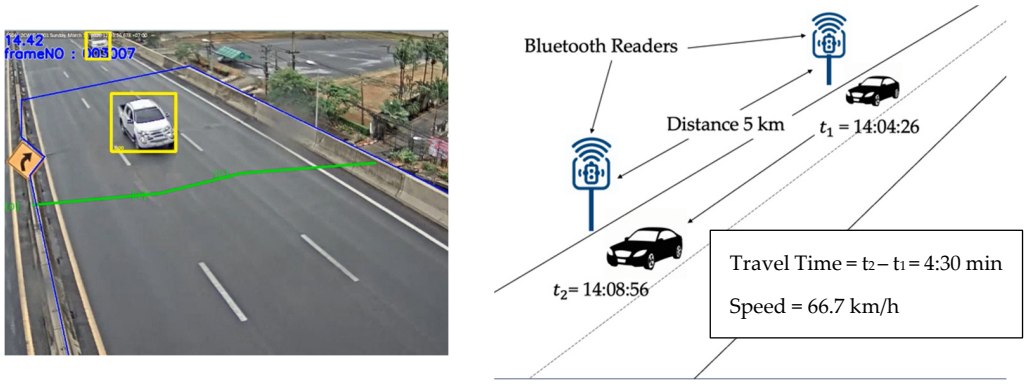


(**a**) (**b**)

Figure 1. Don Muang Tollway study corridor: (**a**) corridor location and (**b**) on-ramp (red arrows) and off-ramp (blue arrows) locations along the corridor.

3.1. Data Collection

In this study, video image processing cameras were used to collect traffic parameters, such as speed and volume, which were aggregated into 5 min intervals. Figure 2a illustrates the beginning and ending screen lines used to calculate the speed of objects inside the boxes. Data regarding travel time ground truth or travel time direct measurement were collected from Bluetooth readers installed along the corridor. Travel time ground truth was calculated by matching the detections of the same MAC address (the same vehicle) at the origin and destination Bluetooth readers [3], as shown in Figure 2b. The Bluetooth devices were installed temporarily and only to collect the training dataset (actual travel times) for travel time model development. As a result, real-time Bluetooth data were not available when the models were deployed. The video image processing software used in this study has vehicle counting and speed measurement accuracy of 80–85 percent.



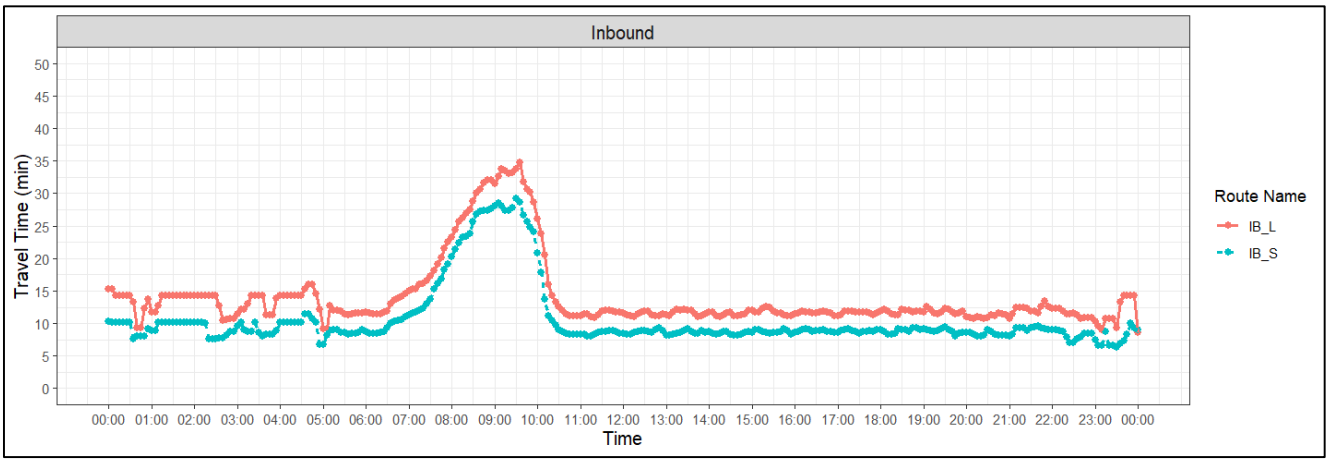
(**a**) (**b**)

Figure 2. Details of data collection: (**a**) speed and volume collected from video image processing systems; (**b**) travel time collected from ground truthing.

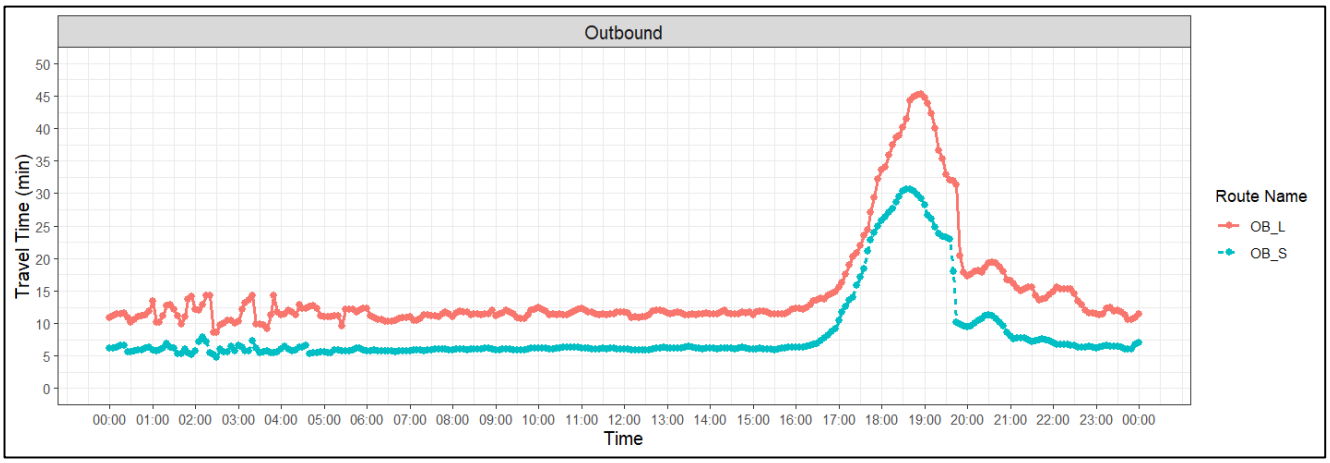
3.2. Study Routes

In total, 11 on-ramps and 13 off-ramps occur along the corridor. The Tollway has developed a system that estimates travel times on several routes every 5 min. We selected four major routes in a variety of directions and lengths, namely, outbound/long distance (OB_L), outbound/short distance (OB_S), inbound/long distance (IB_L), and inbound/short distance (IB_S). Figure 3 illustrates typical travel time characteristics estimated from the Bluetooth readers installed along the study corridor. Route descriptions and locations are shown in Table 1 and Figure 4, respectively. The inbound routes are usually congested from 07:00–10:00, as shown in Figure 3a and the outbound routes are congested from 17:00–20:00, as shown in Figure 3b.

Both the traffic sensor data and Bluetooth data were collected between September and November 2020. The dataset, comprising data from 82 days, was partitioned into distinct training and test sets with an 80/20 division ratio. The first 66 consecutive days of data were used to train the LSTM and SVR models, with approximately 761,280 records (80 percent), each representing a 5-min period over which data were aggregated. Then, the final 16 days of data were used as a test dataset, with about 185,679 records (20 percent), to evaluate model performance. Within these two datasets, 95,160 values (equivalent to about 10 percent) were missing.



(**a**) Typical travel time for inbound routes: inbound/long distance (IB_L) and inbound/short distance (IB_S).



(**b**) Typical travel time for outbound routes: outbound/long route (OB_L) and outbound/short route (OB_S).

Figure 3. Examples of travel times by routes in the study corridor.

Table 1. Descriptions of study routes.

Route Code	Direction	Origin	Destination	Length (km)	No. of VIPSs	Mean Travel Time (min)	Standard Deviation of Travel Time (min)
OB_L	Outbound	Din Daeng	Anusorn Sathan	19.1	68	12.01	1.99
OB_S	Outbound	Din Daeng	Cheangwattana	10.6	40	6.41	1.26
IB_L	Inbound	Anusorn Sathan	Din Daeng	18.9	70	12.21	2.13
IB_S	Inbound	Anusorn Sathan	Lat Prao	13.6	47	8.87	1.65

Note: VIPS denoted video image processing system.

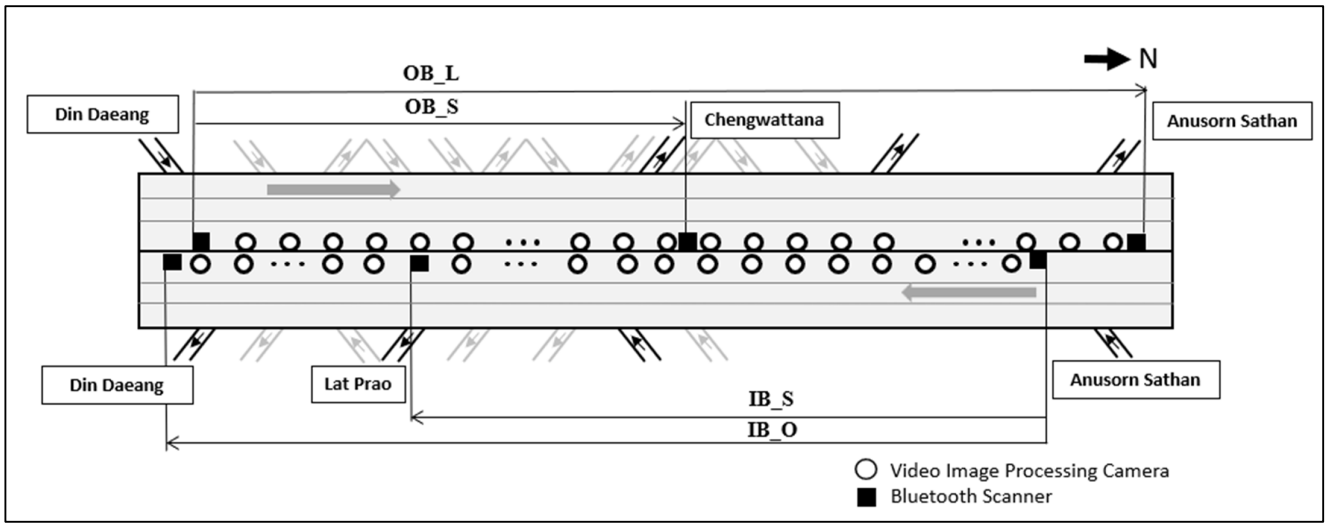


Figure 4. Road segment diagram. Note: IB_L denotes inbound/long distance, IB_S denotes inbound/short distance, IB_L denotes inbound/long distance, and IB_S denotes inbound/short distance.

Notably, only data between 06:00 and 22:00 local time were included for model development. Traffic outside this time window was deemed to be low, allowing the reasonable assumption of free-flow traffic conditions. Furthermore, Bluetooth data volume is rather low during nighttime. As a result, Bluetooth travel time cannot be well estimated in every time slot. Therefore, we excluded low traffic volume from the model.

We also noted that the penetration rate (the ratio between the number of Bluetooth devices and the total number of passing vehicles) on this toll road was between 50 and 60 percent [35]. This figure is substantially higher than the hourly sampling threshold of 2 to 3.4 percent on freeways, suggested by other studies [3,4].

4. Methodology

This section explains how the traffic data and other parameters were handled prior to use in the model development. The overall framework of this research is shown in Figure 5. The process contains data imputation, training and test dataset preparation, model development, and model evaluation.

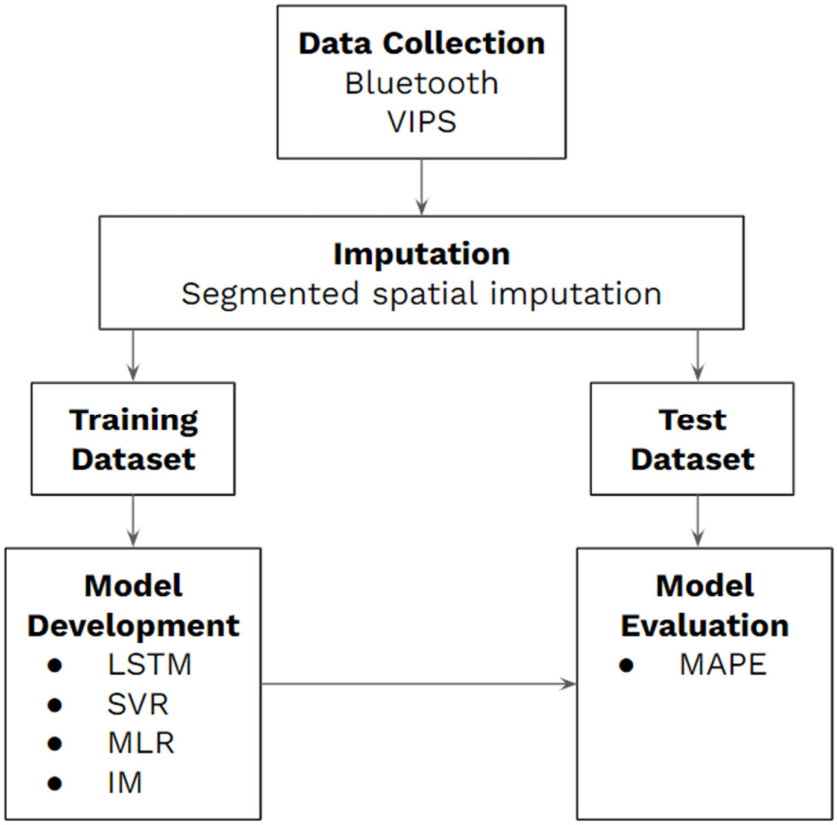


Figure 5. Overview of data processing.

4.1. Data Cleaning and Imputation

Before using the traffic data to develop the travel time models, it was important to properly deal with anomalous data and missing data. In this study, outlier filtering for traffic volume and average speed is conducted using median absolute deviation (MAD) as illustrated in Equations (1) and (2).

$$MAD = \frac{1}{n}\sum_{i=1}^{n} b \cdot |x_i - m(x)| \quad (1)$$

$$T_{min}, T_{max} = m(x) \pm \alpha \cdot MAD \quad (2)$$

where $m(x)$ is the median value of the dataset; n is the number of data values; x_i is the data values in the set; and b is a constant scalar factor. T_{min} and T_{max} are the lower and upper thresholds, respectively, and α is a constant multiplier.

In this study, the b value is set to 1.4826 to make MAD approximately equal to the standard deviation of a normally distributed random variable [35]. After obtaining the MAD, we need to determine the threshold for outlier detection (T_{min}, T_{max}). A widely accepted heuristic is to identify outliers as data points that exceed a certain number of MADs from the median. The threshold can be set based on a constant multiplier, typically denoted as α. In this study, we selected the α value to be equal to 3, following the approach used in a previous study [35,36]. Data points that are further away from these values are considered outliers. Data points identified as outliers, comprising approximately 10% of the dataset, were subsequently removed.

Across the entire 82 days of data, 4 to 24 percent of the data were missing. When missing data were found, we used the segment imputation described in this section to handle the missing data.

Other approaches [18,37] used the spatial correlation between traffic data at adjacent detector stations, with the spatial interpolation imputing the missing traffic data at a

detector station by replacing it with the average value of the measurements obtained from the adjacent upstream and downstream stations. For a detector on a northbound highway, its upstream station is the one immediately to its north, and its downstream station is directly to its south. However, if the adjacent detector also does not have collected data, the missing data cannot be imputed.

In this study, we developed a novel spatial imputation method to choose the nearest detectors from upstream and downstream in the same segment and period to replace the missing data. We divided the segments by entrance/exit ramp locations along a tollway. The spatial trends were used to correct missing values in the same segments with similar traffic patterns under the assumption that the traffic volume and speed of data collected from detectors in the same segments influenced each other. There are two cases considered using this imputed method:

Case 1: If upstream and downstream detectors in the same segment are available, the average value from the nearest upstream and downstream detectors will be used to fill in the missing value, as shown in Figure 6a.

Device ID	Segment ID	Time	Speed (km/hr)
CAM001	1	06:00	110.1
CAM002	1	06:00	89.1
CAM003	1	06:00	118.1
CAM004	1	06:00	*NA*
CAM005	1	06:00	*NA*
CAM006	1	06:00	94.8



Device ID	Segment ID	Time	Speed (km/hr)
CAM001	1	06:00	110.1
CAM002	1	06:00	89.1
CAM003	1	06:00	118.1
CAM004	1	06:00	106.4
CAM005	1	06:00	106.4
CAM006	1	06:00	94.8

(**a**) Case 1: Both upstream and downstream traffic data are available.

Device ID	Segment ID	Time	Speed (km/hr)
CAM001	1	06:00	*NA*
CAM002	1	06:00	*NA*
CAM003	1	06:00	*NA*
CAM004	1	06:00	*NA*
CAM005	1	06:00	*NA*
CAM006	1	06:00	94.8



Device ID	Segment ID	Time	Speed (km/hr)
CAM001	1	06:00	94.8
CAM002	1	06:00	94.8
CAM003	1	06:00	94.8
CAM004	1	06:00	94.8
CAM005	1	06:00	94.8
CAM006	1	06:00	94.8

(**b**) Case 2: Only one piece of traffic data is available in a segment.

Figure 6. Segment imputation scenarios.

Case 2: If only one detector in the same segment is available, the value from the active detector will be used to fill in the missing value for the other sensors within the same segment, as shown in Figure 6b.

By using this approach, approximately 90 percent (85,644 values) of the missing data were imputed. For each of the remaining missing data (9516 records), a value of zero was assigned. To differentiate an actual zero value from a missing value, another input variable called "missing data indicator" was created, with a value of 1 for missing data and 0 otherwise [8].

4.2. Attribute Coding

Two types of variables were used for model development: continuous variables and categorical variables. Continuous variables, such as speed and volume at 5 min

intervals, were collected from VIPSs installed every 100 to 500 m along the corridor. Data normalization was required before these continuous variables could be used as inputs in the machine learning model, as discussed in Section 4.3. The categorical variables used were (i) time of day, (ii) day of the week, (iii) holiday indicator, (iv) day-before-holiday indicator, and (v) day-after-holiday indicator. A dummy coding method was used to encode these categorical variables ($c - 1$ variables for the c groups). The data attributes for all variables are shown in Table 2.

Table 2. Data attributes.

Variables	Description	Variable Type	Example
AVGSPEED_ [CAMERAID]	Average speed (km/h) at 1 min intervals	Continuous	AVGSPEED_007, AVGSPEED_008, . . ., AVGSPEED_184
TOTALCOUNTS_ [CAMERAID]	Vehicle count (veh/min)	Continuous	TOTALCOUNTS_ 007, TOTALCOUNTS_ 008 . . ., TOTALCOUNTS_184
MISSDATAIND_ [CAMERAID]	Missing data indicator	Dummy	1: Missing Data 0: Otherwise
DOW1 DOW2 DOW3 DOW4 DOW5 DOW6	Day of week	Dummy	{DOW1 = 0, DOW2 = 0, . . ., DOW6 = 0} for Sunday; {DOW1 = 1, DOW2 = 0, . . ., DOW6 = 0} for Monday; {DOW1 = 0, DOW2 = 1, . . ., DOW6 = 0} for Tuesday; . . .
HR06 HR07 HR08 HR23	Time of day	Dummy	{HR07 = 0, HR08 = 0, . . ., HR23 = 0} for 06:00–06:59; {HR07 = 1, HR08 = 0, . . ., HR23 = 0} for 07:00–07:59; {HR07 = 0, HR08 = 1, . . ., HR23 = 0} for 08:00–08:59; . . .
HOLIDAY	Holiday indicator	Dummy	HOLIDAY = 1 if holiday; HOLIDAY = 0 otherwise
B_HOLIDAY	Day-before-holiday indicator	Dummy	B_HOLIDAY = 1 if before the holiday; B_HOLIDAY = 0 otherwise
A_HOLIDAY	Day-after-holiday indicator	Dummy	A_HOLIDAY = 1 if after the holiday; A_HOLIDAY = 0 otherwise

4.3. Data Normalization

Since speed and volume data have different magnitudes and units, data normalization is required to convert the data into a narrower and more specific range [9]. Means and standard deviations were calculated from the training dataset and were then entered in Equations (3) and (4) for both the training and test datasets to obtain a normalized dataset [10].

$$Z_{train} = \frac{x_{train} - \mu_{train}}{\sigma_{train}} \tag{3}$$

$$Z_{test} = \frac{x_{test} - \mu_{train}}{\sigma_{train}} \tag{4}$$

where z is the standardized value; x is the observation of the feature, such as speed or volume; μ is the mean value for the feature, and σ is the standard deviation value for the feature.

5. Model Development

In this study, two models based on machine learning technology were developed to estimate travel time in the study corridor: SVR and LSTM neural networks. Simple models,

consisting of multiple linear regression (MLR) and an instantaneous model (IM), were also constructed for use as baselines to compare with the machine learning models.

5.1. Support Vector Regression

A support vector machine is a supervised machine learning method developed by Cortes and Vapnik in 1995 for use in binary classification [38]. It has since been developed for use in regression, in which context it is called SVR [39]. In the domain of machine learning, SVR is considered an effective and efficient algorithm and has been extensively studied. SVR-based traffic methods have been proven to outperform time series and classical regression methods in several studies [30,40].

The basic idea of SVR is to map the training data from the input space into a higher-dimensional feature space via an appropriate function. Then, a hyperplane whose margin in the feature space is maximized can be constructed to separate the data. Thus, nonlinear regression problems are handled by a set of nonlinear transfer functions that map the input space into a high-dimensional feature space. According to statistical learning theory, the linear estimation function of SVR can be formulated using Equation (5):

$$f(x) = \omega \cdot \phi(x) + b \tag{5}$$

where $\varnothing(x)$ denotes a nonlinear transfer function in the feature space, ω is the weight vector, and b is a constant. The coefficients ω and b can be calculated by minimizing the risk function shown in Equation (6):

$$\sum_{i=1}^{n} L_\varepsilon(y_i, f(x_i)) + \frac{1}{2}||w||^2 \tag{6}$$

$$L_\varepsilon = (y, f(x)) = \begin{cases} 0, \ if(y - f(x)) \le \varepsilon \\ |y - f(x)| - \varepsilon, \ otherwise \end{cases}$$

$L_\varepsilon(y, f(x))$ is called the ε-intensive loss function, where ε is the tube size that is equivalent to the approximation accuracy placed on the training points. The constant $C > 0$ specifies a trade-off between the approximation error and the weight vector $||w||$. Both C and ε must be chosen beforehand by the user.

Two non-negative slack variables ξ and ξ^* can be introduced, representing the distances from the actual values to the corresponding boundary values of the ε-tube. Thus, Equation (6) can be transformed into the convex quadratic programming problem expressed using Equation (7):

$$min_{w,b,\xi,\xi^*} \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{7}$$

$$subject\ to \begin{cases} w_i \cdot \phi(x_i) + b_i - y_i \le \varepsilon + \xi_i^* \\ y_i - w_i \cdot \phi(x_i) - b_i \le \varepsilon + \xi_i \\ \xi_i, \xi_i^* \ge 0, \ i = 1, 2, \ldots, N \end{cases}$$

After optimizing the above problem via a Lagrange function and appropriate conditions, the nonlinear regression function can be obtained, as shown in Equations (8) and (9):

$$f(x) = \sum_{i=1}^{l}(a_i - a_i^*)k(x_i, x) + b \tag{8}$$

$$k(x_i, x_j) = \exp(-\gamma||x_i - x_j||) + b \tag{9}$$

where α_i and a_i^* are two Lagrange multipliers, and $k(x_i, x_j)$ is a radial basis function (RBF) that describes the inner products in the high-dimensional feature space.

The performance and efficiency of an SVR depend greatly on the kernel function. Therefore, it is imperative to properly choose the kernel function and corresponding

parameters according to the problem at hand. The three common kernel functions are linear, RBF, and sigmoid [41]. In the present study, grid search was utilized to compare the model performance among the three kernel functions, including linear, RBF, and sigmoid. The results demonstrated that the RBF function had the highest performance (i.e., lowest MAPE) compared to the other functions. Therefore, we chose RBF as the kernel function. RFB has also been widely used in previous travel time studies [30,40]. Equation (9) represents the RBF employed, while Figure 7 illustrates the SVR structure.

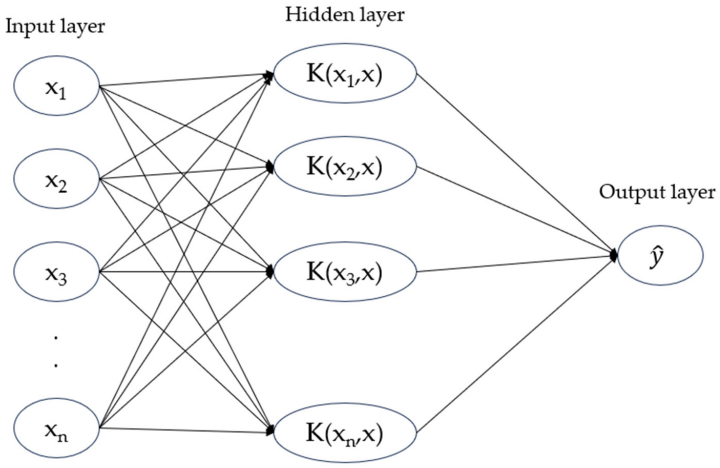


Figure 7. Structure of support vector regression.

5.2. Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are commonly used with sequential data, such as handwriting recognition, speech recognition, and time-series data. Unlike the output of a traditional neural network, an RNN depends not only on the current input but also on the previous state of the network, which serves as memory.

In this research, we used LSTM architecture for our recurrent neural networks [42]. LSTM remembers long-term information differently by computing the hidden state of the network. The hidden state of an LSTM contains a sequence of memory blocks with special gates to control how long the information is kept in each memory block. The basic LSTM architecture consists of three main layers: input, LSTM, and output. The main improvement of LSTM cells over simple recurrent cells is the addition of forget gates. As shown in Figure 8, an LSTM cell has a forget gate and an update gate. The forget gate allows an LSTM cell to remove or add needed information to the cell state [33]. Other studies [9,32,33,43,44] have shown that LSTM-based traffic estimation methods outperformed time series and classical regression methods.

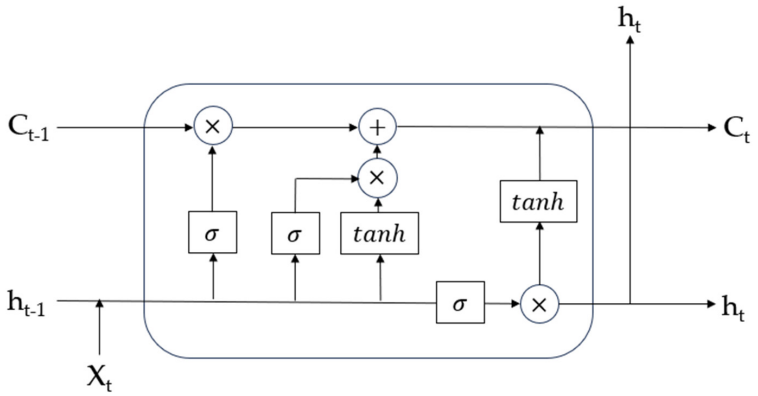


Figure 8. Structure of long short-term memory cells. Note: X denotes time-series data, h_t denotes hidden vector, C_t denotes cell state, σ and tanh denotes activation functions, f_t denotes forget gate, i_t denotes input gate, and o_t denotes output gate.

Let the input time-series data in the sliding window be $X = \{x_1, x_2, \ldots, x_T\}$. Then, the hidden vector sequence $H = \{h_1, h_2, \ldots, h_T\}$ is calculated. Next, an output sequence $Y = \{y_1, y_2, \ldots, y_T\}$ is given by the LSTM network sequence. Then, Equations (10) and (11) are iterated:

$$h_t = H(W_h \cdot [h_{t-1}, x_t] + b_h) \tag{10}$$

$$y_t = W_y h_t + b_y \tag{11}$$

where W is a weight matrix (for example, W_h is the input-hidden weight matrix) and b is a bias vector. $H(\cdot)$ is a hidden layer function and is computed by iterating Equations (12)–(17):

Gates:

$$f_t = \sigma\Big(W_f \cdot [h_{t-1}, x_t] + b_f\Big) \tag{12}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x] + b_i) \tag{13}$$

$$\widetilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \tag{14}$$

Input transform:

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \widetilde{C}_t \tag{15}$$

Memory update:

$$O_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{16}$$

$$h_t = O_t \cdot tanh(C_t) \tag{17}$$

where $\sigma(\cdot)$ is a sigmoid function, $\sigma(\cdot)$ and $\tan h(\cdot)$ are defined in Equations (16) and (17), respectively, f is a forget gate, i is an input gate, o is an output gate, and C is a cell update gate. Equations (18) and (19) together define the activate function.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{18}$$

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{19}$$

5.3. Model Performance Measurement

In this study, the performance evaluation of each estimation method was based on common indicators: the mean absolute percentage error (MAPE), the mean absolute error (MAE), and the root mean squared error (RMSE). MAPE highlights the relative accuracy and percentage deviation of estimations. MAE provides the average magnitude of errors, regardless of their directions. RMSE emphasizes the influence of large errors and indicates the spread of errors. The three metrics are defined in Equations (20) and (21):

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{t_i - \hat{t}_i}{t_i} \right| x100 \tag{20}$$

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{t_i - \hat{t}_i}{t_i} \right| \tag{21}$$

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^{n} ||t_i - \hat{t}_i||^2}{n}} \tag{22}$$

where n is the number of testing samples, t_i is an observed travel time value, and $\hat{t}$ indicates an estimated travel time value output by the estimation method. The standard

interpretation of MAPE for travel time estimation models is described in [45]. In the context of this matter, MAPE values of less than 10 percent are categorized as "highly accurate", while MAPE values ranging between 10 and 20 percent are characterized as "good". Additionally, if the MAPE values fall within the range of 20 to 50 percent, they are classified as "reasonable", and any value below 50 percent is designated as "inaccurate".

5.4. Hyperparameter Calibration and Optimization

In this section, we describe the hyperparameter tuning and optimization. For SVR models, kernel function selection is also an important factor that influences performance. The RBF is an adaptive kernel function for low-dimensional space data with a good convergence domain in high-dimensional space. RBF is normally used for regression [30,34,46]. For this kernel function, three parameters must be optimized: ε, cost I, and γ. A value of 0.1 was assigned to ε, as used elsewhere [40]. The other two parameters were optimized using the grid search method with fivefold cross-validation.

For the LSTM model, the three parameters to be optimized were (i) learning rate, (ii) epoch, and (iii) batch size. Based on other studies [9,10,47], we selected 128 as the batch size and 2 as the number of hidden layers. Then, the epoch value was optimized for each of the four models, as shown in Table 3. To determine the optimal number of backward timesteps, a sensitivity analysis of various backward timesteps (1, 5, 10, 15, 30, and 60 min) was conducted to find the model with the minimum MAPE. Figure 9 shows that the 15-min timestep model had the lowest MAPE and therefore was chosen for the final LSTM model.

Table 3. The hyperparameters tuning and optimization of each model.

Parameters	*OB_L*	*OB_S*	*IB_L*	*IB_S*
SVR				
ε	0.1	0.1	0.1	0.1
Cost	10	1000	1000	1
γ	0.001	0.0001	0.0001	0.001
LSTM				
Learning rate	0.01	0.01	0.01	0.01
Batch size	128	128	128	128
Epoch	45	20	65	65

Note: SVR denotes support vector regression, LSTM denotes long short-term memory, OB_L denotes outbound long route, OB_S denotes outbound short route, IB_L denotes inbound long route, and IB_S denotes inbound short route.

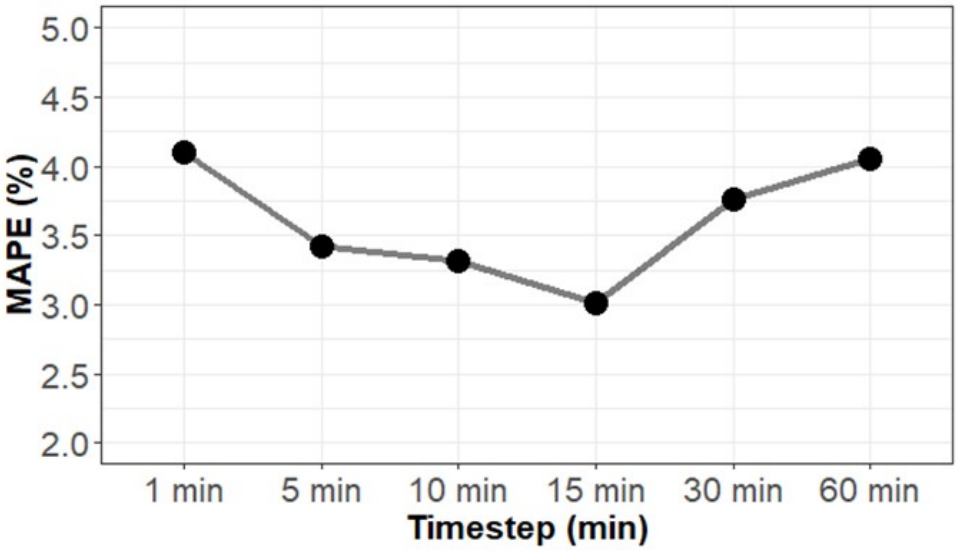


Figure 9. Sensitivity analysis of backward timesteps for long short-term memory model. Note: MAPE denotes the mean absolute percentage error.

6. Results

For model validation, we compared the travel time estimation performance of the two machine learning methods (SVR and LSTM) with other commonly used methods: multiple linear regression (MLR) and an instantaneous model (IM).

The link travel time at time t under IM can be formulated using Equation (23):

$$y(t) = \frac{D}{\left(\frac{v_o\ (t)+v_d\ (t)}{2}\right)} \tag{23}$$

where D is the length (km) of a given route, and $v_o\ (t)$ and $v_d\ (t)$ denote the average speeds (at 5-min aggregation, km/h) upstream and downstream, respectively, of a link at time t.

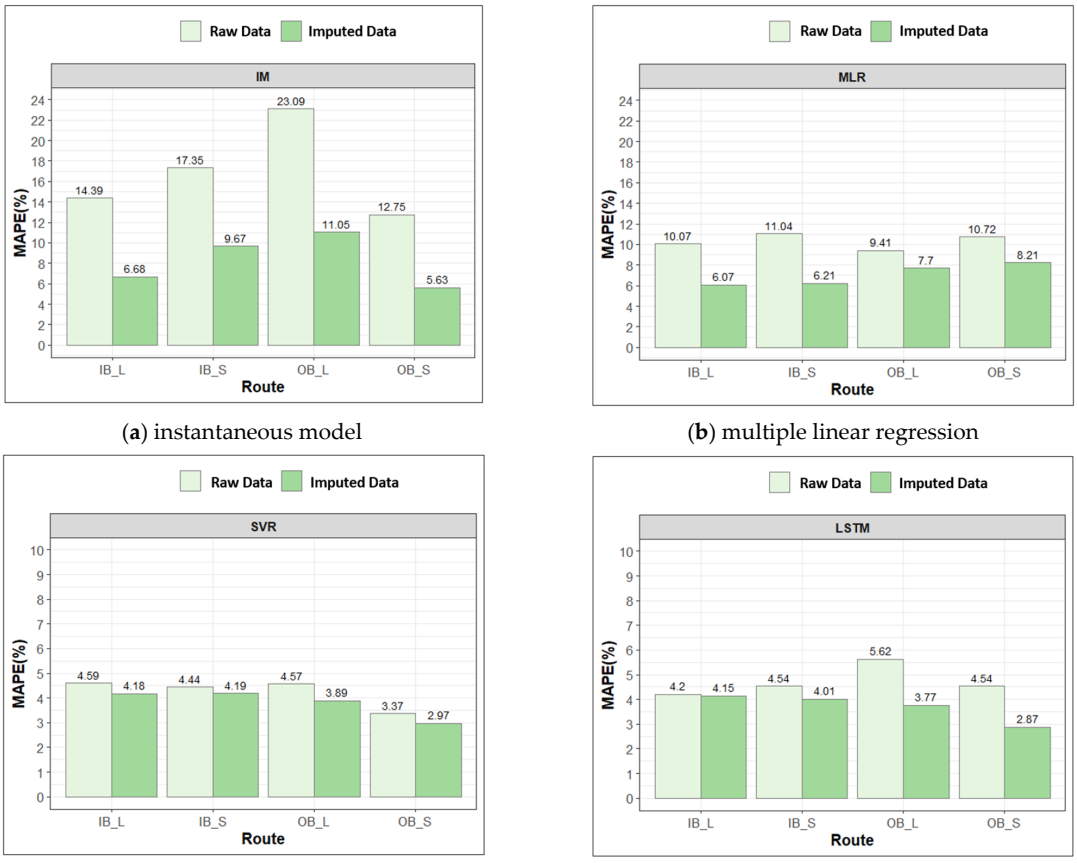
The link travel time between the origin and destination under MLR can be expressed using Equation (24):

$$y(t) = \sum_{i=1}^{n} \beta_i \cdot x_i\ (t) \tag{24}$$

where β_i is the regression coefficient of the ith camera, x_i is the travel time (seconds) to reach the ith camera, n is the number of image processing cameras between the origin and destination, and t is the time index.

6.1. Effects of Data Imputation

The MAPEs of the four travel time models—instantaneous model (IM), multiple linear regression (MLR), long short-term memory (LSTM), and support vector regression (SVR)—are summarized in Figure 10 and Table 4, as are the effects of data imputation on the model performance.



(**a**) instantaneous model (**b**) multiple linear regression

(**c**) support vector regression (**d**) long short-term memory (LSTM)

Figure 10. MAPE results of four travel time models: (**a**) instantaneous model (IM), (**b**) multiple linear regression (MLR), (**c**) support vector regression (SVR), and (**d**) long short-term memory (LSTM).

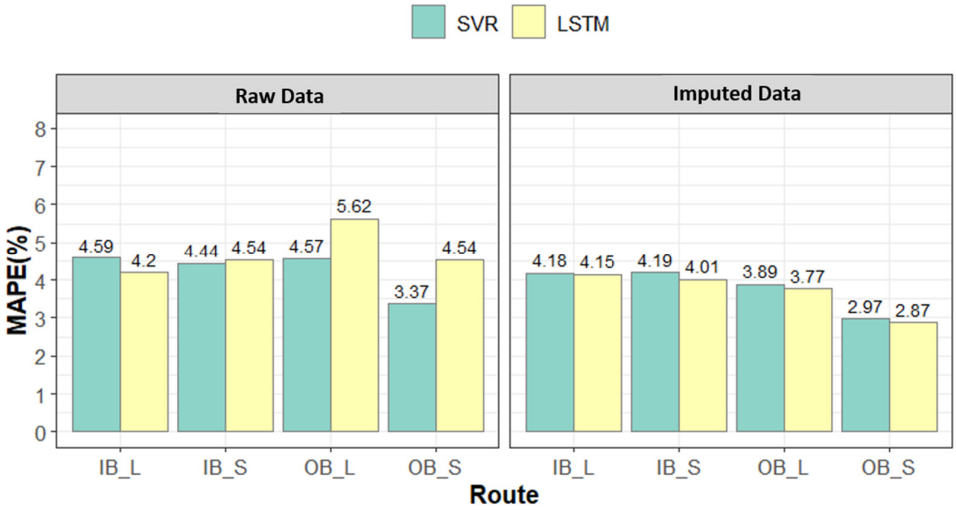
Table 4. Weekday travel time estimation results for four routes (OB_S, OB_L, IB_S, IB_L) based on four modeling methods: instantaneous model (IM), multiple linear regression (MLR), support vector regression (SVR), and long short-term memory (LSTM).

Route	Distance(km)	Period	MAPE (%)				MAE (min)				RMSE (min)			
			IM	MLR	SVR	LSTM	IM	MLR	SVR	LSTM	IM	MLR	SVR	LSTM
IB_L	18.9	AM	11.27	14.18	7.64	7.37	2.00	2.92	1.77	1.32	3.34	5.98	4.91	2.57
		MD	3.14	2.92	3.42	2.97	0.37	0.34	0.39	0.34	0.47	0.43	0.50	0.43
		PM	6.07	3.45	3.07	2.81	0.74	0.42	0.37	0.34	0.84	0.54	0.45	0.43
		ALL	6.68	6.07	4.18	4.15	1.48	1.24	0.69	0.55	2.53	2.69	2.09	1.10
IB_S	13.6	AM	14.09	13.29	5.90	3.94	1.78	2.19	0.92	0.46	2.58	4.57	2.13	0.65
		MD	5.77	4.08	3.01	2.76	0.50	0.35	0.26	0.24	0.57	0.43	0.33	0.29
		PM	10.18	4.20	3.72	4.19	0.93	0.38	0.33	0.38	1.01	0.46	0.42	0.45
		ALL	9.67	6.21	4.19	4.01	1.37	0.96	0.47	0.40	2.03	1.99	1.21	0.63
OB_L	19.1	AM	11.30	6.79	4.17	4.44	1.38	0.81	0.52	0.56	1.66	1.08	0.88	0.88
		MD	9.36	5.68	3.60	3.50	1.11	0.67	0.42	0.41	1.22	0.83	0.53	0.52
		PM	18.33	11.30	5.91	5.28	2.67	1.62	0.74	0.78	3.09	2.04	1.19	1.04
		ALL	11.05	7.70	3.89	3.77	2.30	1.18	0.63	0.56	3.36	2.01	1.09	0.92
OB_S	10.6	AM	4.04	6.09	2.82	2.46	0.25	0.38	0.18	0.16	0.33	0.49	0.31	0.26
		MD	2.86	5.27	1.99	2.28	0.18	0.33	0.12	0.14	0.22	0.41	0.15	0.17
		PM	11.07	15.97	5.44	3.84	0.94	1.33	0.47	0.32	1.31	1.81	0.75	0.49
		ALL	5.63	8.21	2.97	2.87	0.70	0.72	0.25	0.23	1.22	1.27	0.48	0.48

Note: AM denotes morning, MD denotes midday, and PM denotes evening periods, and ALL denotes 06:00–22:00. Each grey-shaded cell highlights the best performing model, i.e., smallest error, for each period on each route.

The light green bars shown in Figure 10 indicate model performance without data imputation. Clearly, the MAPEs of SVR are the lowest. In particular, the MAPE values of SVR and LSTM are noticeably lower than those of the baseline models (IM and MLR).

The dark green bars in Figure 10 indicate the model performance using the imputed data. Overall, the model performance improved with data imputation. Additionally, the MAPEs substantially decreased, especially for the baseline models (IM and MLR), as shown in Figure 10a,b, respectively. However, the imputed data improved the performance of the machine learning models (LSTM and SVR) to a lesser degree, as shown in Figure 10c,d, respectively. Figure 11 compares the LSTM and the SVR model performance levels with and without data imputation. It can be seen that without data imputation, the MAPEs of the SVR were higher than those of the LSTM models. Furthermore, with the imputed data, the MAPEs of the LSTM were higher than those of the SVR model.



(**a**) Raw data (**b**) Imputed data

Figure 11. MAPE results of SVR and LSTM models.

In addition, a shorter route generally resulted in lower MAPEs for any given model and period, indicating that shorter routes had greater accuracy in travel time estimation.

Various travel time patterns were considered according to the time of day by comparing the results across three different periods: morning (07:30–08:30), midday (12:00–13:00), and evening (17:30–18:30). Traffic volume peaks are common during the morning period for inbound traffic and in evening periods for outbound traffic. Consequently, the travel time in these periods fluctuated substantially. However, during the midday period, travel times remained comparatively stable.

6.2. Model Performance under Various Traffic Conditions

Table 5 summarizes model performance in terms of three common metrics, including MAPE, MAE, and RMSE, during weekdays. Each grey-shaded cell highlights the model with the best performance (the least MAPE for a given route). Unsurprisingly, MAPE values, along with the other metrics for SVR and LSTM, were substantially lower than those of the baseline models (IM and MLR). Additionally, the MAPE values for the SVR and LSTM models were less than 10%, indicating that these models are highly accurate based on the interpretation in [45].

Table 5. Model performance during peak periods of days before/after long holidays.

Route	Period	MAPE (%)				MAE (min)				RMSE (min)			
		IM	**MLR**	**SVR**	**LSTM**	**IM**	**MLR**	**SVR**	**LSTM**	**IM**	**MLR**	**SVR**	**LSTM**
IB_L	2020-11-2307:30–09:30	35.27	49.56	40.36	20.14	7.34	17.38	13.87	6.56	9.08	18.63	16.12	7.93
IB_S	2020-11-2307:30–09:30	18.29	46.67	23.49	8.10	4.78	13.63	5.95	1.24	5.90	14.19	6.80	1.45
OB_L	2020-11-1818:00–19:30	26.43	17.23	13.05	5.81	3.31	2.42	1.18	1.15	3.75	3.36	1.63	1.44
OB_S	2022-11-1818:00–19:30	16.03	27.90	10.41	7.78	2.35	3.86	1.15	1.04	3.29	4.40	1.32	1.24

Note: Each grey-shaded cell highlights the model with the best performance, i.e., the smallest errors, for each period on each route.

In terms of MAPE, the LSTM models marginally outperformed the SVR models for congested traffic conditions, such as outbound traffic in the evening (3.84% for LSTM vs. 5.44% for SVR on OB_S route; 5.28% for LSTM vs. 5.91% for SVR on OB_L route) and inbound traffic in the morning (3.94% for LSTM vs. 5.90% for SVR on IB_S route; 7.37% for LSTM vs. 7.64% for SVR on IB_L route). It was apparent that the MAPEs of the LSTM models with shorter lengths were substantially lower.

In contrast, the SVR models performed similarly to LSTM models for lower traffic volume, such as evening off-peak for the inbound direction, morning off-peak for the outbound direction, and midday for both directions. Surprisingly, the MLR model outperformed other models in lower volumes at midday in the inbound direction. Additionally, the MAPE values for the model and period were generally higher during the peak period than during other periods since the turbulence of travel time was generally higher during peak periods.

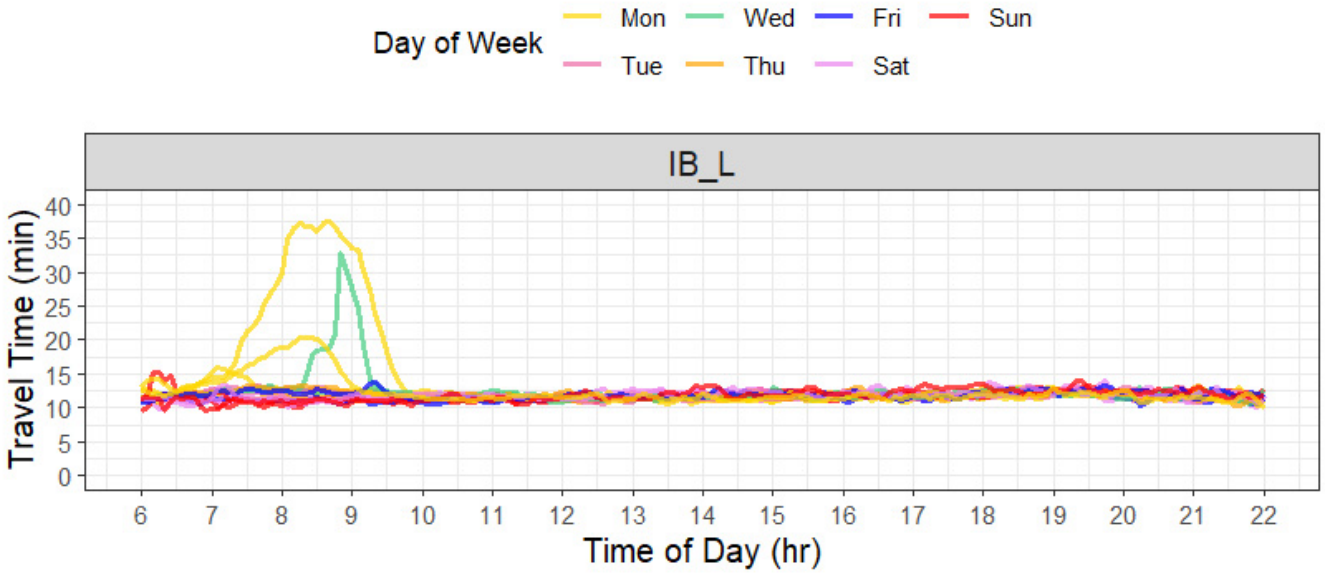
6.3. Model Performance under Unusual Traffic Conditions

In this section, we focus on model performance under traffic congestion, specifically the traffic dataset just before and after a long holiday from 19–22 November 2020. The most congested periods for each direction were expected to occur as follows:

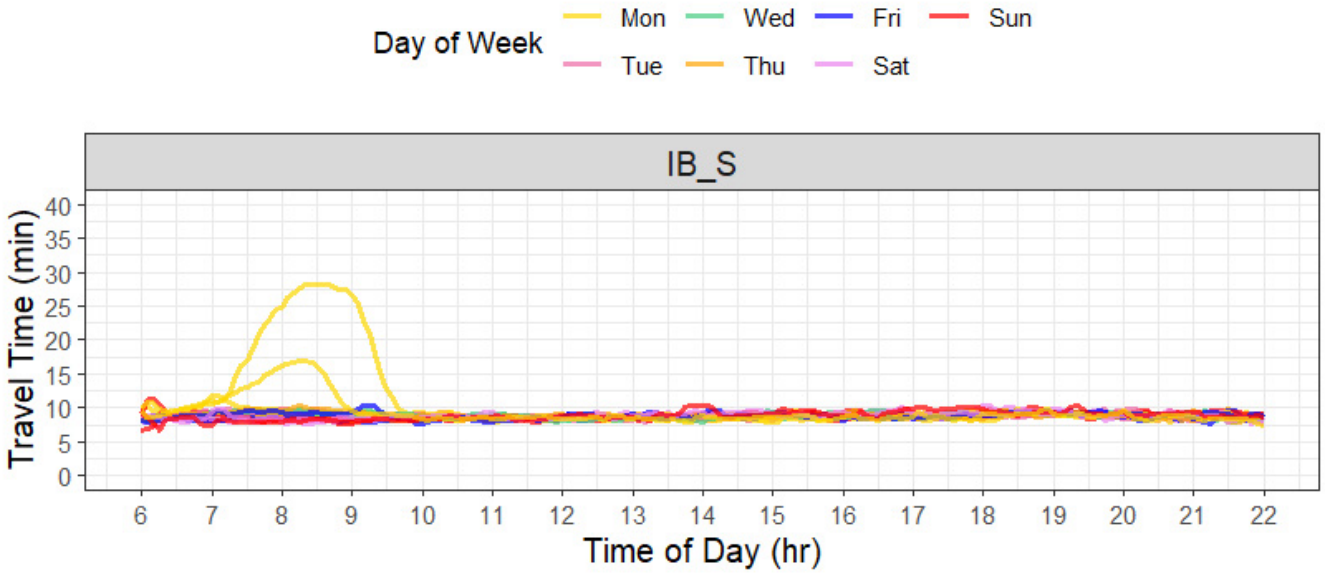
- The inbound direction during the morning rush hour of 23 November 2020 (Monday), which was the first day of work following the long holiday, as shown in Figure 12.
- The outbound direction during the evening of 18 November 2020 (Wednesday), when people left the city for other provinces, as shown in Figure 13.

For the highly congested traffic conditions, the MAPEs of the LSTM models were noticeably lower than those of the SVR models for both the evening before the holiday and the morning following the holiday (see Table 5). In addition, the predicted values from the

four models are shown in Figures 14 and 15. We can see that although there are differences in the model curves, the LSTM model was able to detect the variations with very high accuracy. In short, LSTM clearly outperformed SVR when traffic was unusually high.

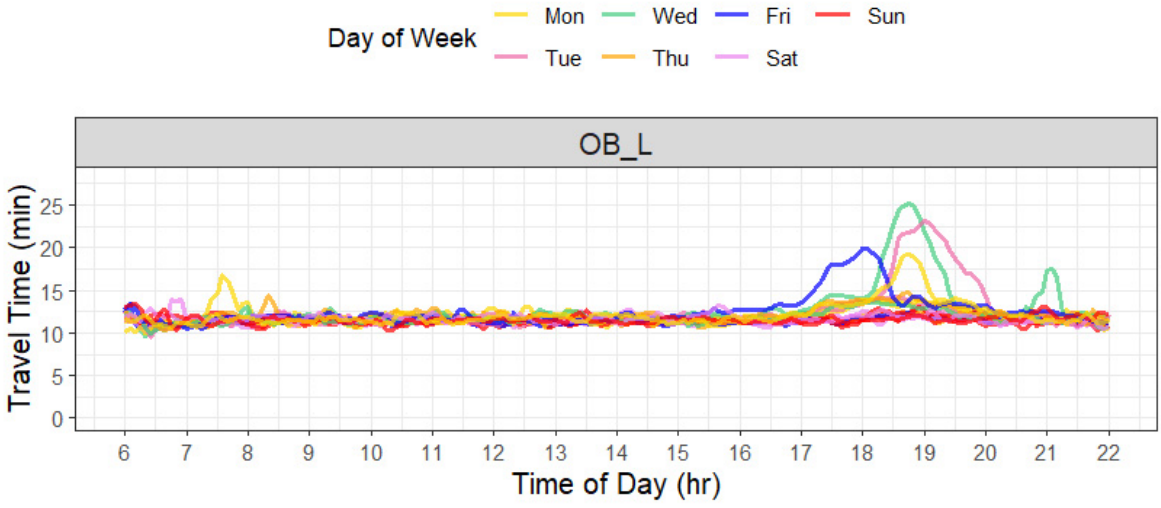


(**a**)

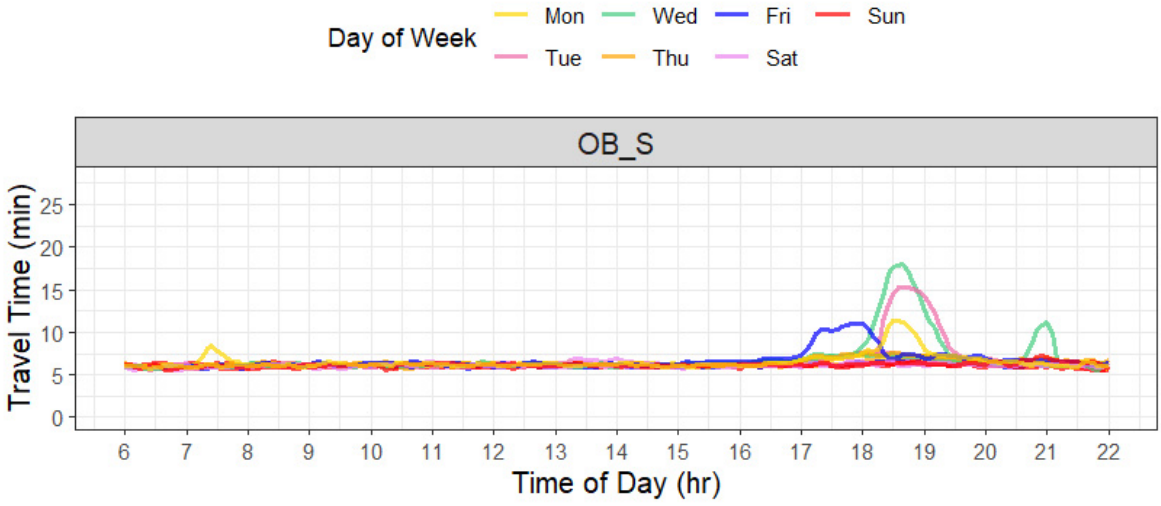


(**b**)

Figure 12. Weekday travel time data collected from Bluetooth readers: (**a**) inbound/long distance (IB_L); (**b**) inbound/short distance (IB_S).

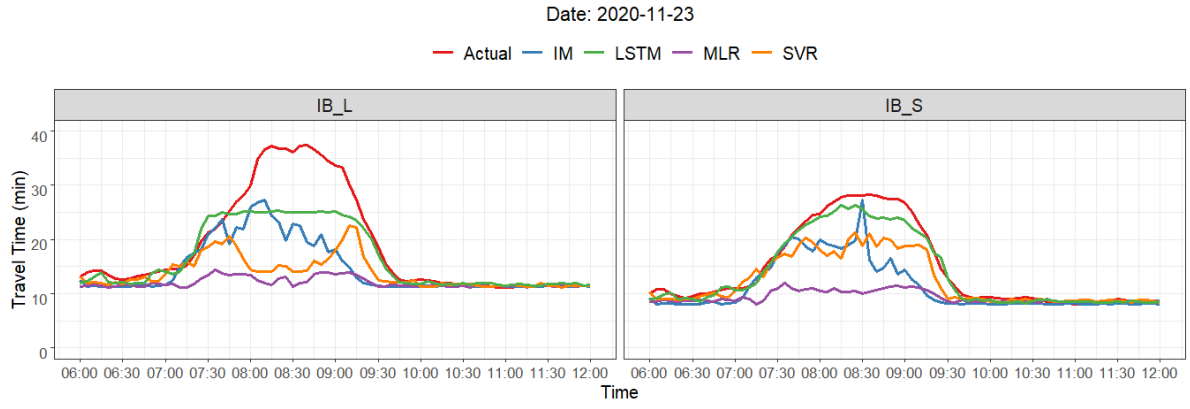


(**a**)



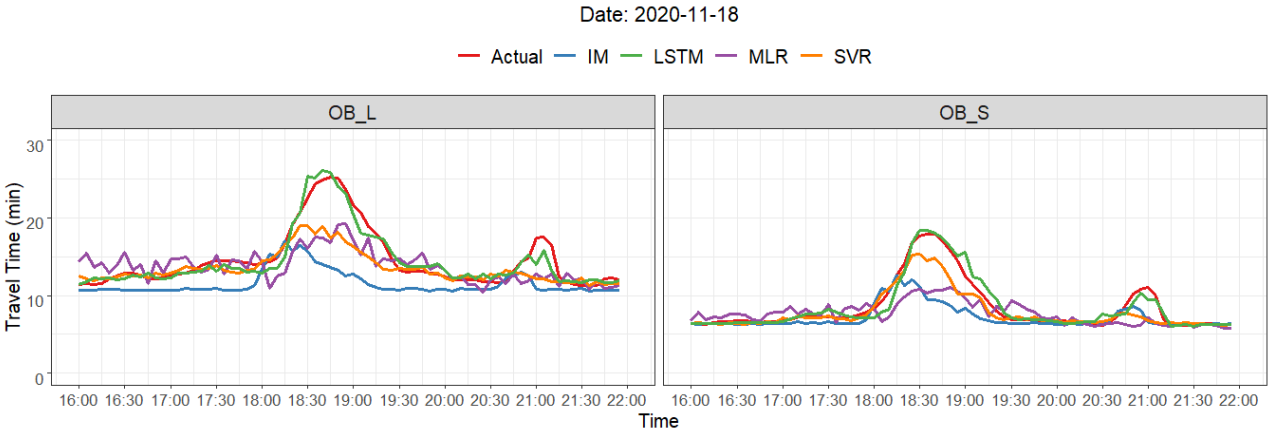
(**b**)

Figure 13. Weekday travel time data collected from Bluetooth readers: (**a**) outbound/long distance (OB_L); (**b**) outbound/short distance (OB_S).



(**a**) (**b**)

Figure 14. Estimated vs. actual travel times during high traffic congestion for the day after a long holiday (**a**) inbound/long distance and (**b**) inbound/short distance on Monday, 23 November 2020.



(**a**) (**b**)

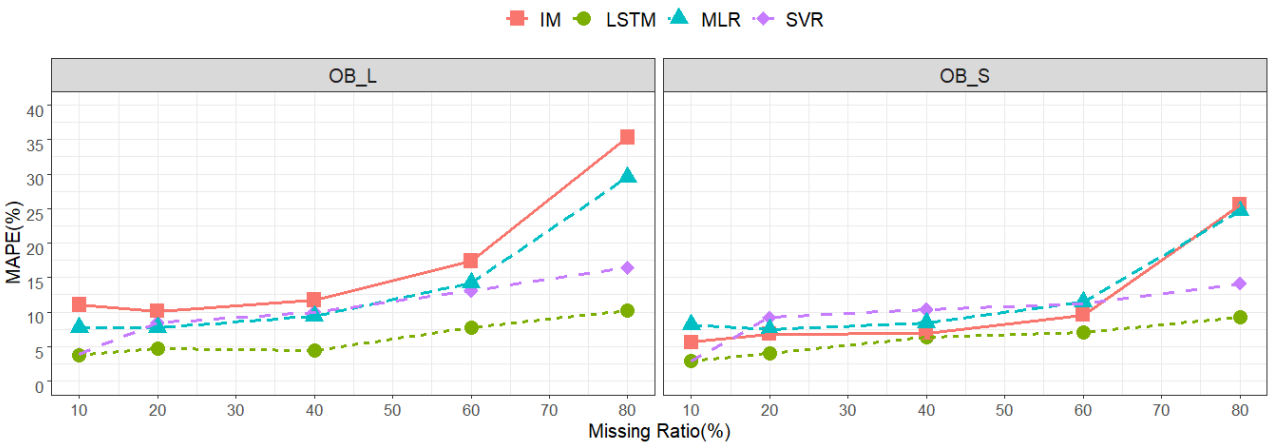
Figure 15. Estimated vs. actual travel times during high traffic congestion for the day before a long holiday (**a**) outbound/long distance and (**b**) outbound/short distance on Wednesday, 18 November 2020.

6.4. Effects of Missing Data on Model Performance

This section determines the sensitivity of the model performance to the missing data. It is known that missing data can affect model performance [5]. However, few studies have evaluated the sensitivity effect of missing data on model performance.

Therefore, we determined how various models were affected by travel time based on generating random ranges from 10% to 80% in increments of 20% (the original datasets had approximately 10% missing data) and replaced the missing values with our segment imputation. We selected outbound direction routes because these routes were often very congested.

Figure 16 shows the results of the model performance evaluation based on simulated missing values ranging from 10 to 80 percent. As the level of missing data increased, the MAPEs of all models increased, but at different rates. For example, the MAPEs of the MLR and IM models increased substantially when the missing data level exceeded 60%, while the MAPEs of the LSTM and SVR models increased relatively slowly. Additionally, the LSTM model had a lower MAPE than the SVR model, especially for a higher range of missing data, implying that the LSTM was more robust to missing data than the SVR model. Notably, the differences in model performance (based on the MAPE values) were less prominent for the short route compared to the longer route.



(**a**) (**b**)

Figure 16. Results of performance evaluation according to missing values in outbound routes: (**a**) outbound/long distance, (**b**) outbound/short distance.

7. Discussion and Conclusions

This paper estimated travel times on a toll road in Bangkok, Thailand, based on two developed machine learning-based models: SVR and LSTM. The mode inputs were traffic parameters, such as speed and flow, that were collected from image processing cameras. Travel time models were constructed for four selected routes (OB_S, OB_L, IB_S, and IB_L) with different lengths and directions. Bluetooth readers were installed along the study corridor to collect the ground truth travel times used in the model calibration and validation analysis. In addition, we developed traditional spatial trends by dividing the tollway into segments. Our spatial trends were used to correct missing values in the same segments with similar traffic patterns under the assumption that the traffic volume and speed data collected from detectors in the same segments influenced each other.

The SVR and LSTM models were compared with two baseline models (IM and MLR), as shown in Figure 10. The machine learning models (SVR and LSTM) performed better than the baseline models for both the raw and imputed input data. Additionally, our imputation method improved the performance of all four models.

The comparison of the SVR and LSTM models (Figure 11) indicated that SVR was more robust to missing raw input data, and this effect was more prominent for higher traffic congestion on the outbound routes (OB_L and OB_S). However, LSTM performed better once the input data had been imputed.

We also considered model performance during heavier traffic periods, such as 6:00–22:00. Table 4 presents the performance comparison of the SVR and LSTM models with the other two models in terms of MAPE, MAE, and RMSE. The results indicate that the SVR and LSTM models outperformed the other models across various traffic conditions and routes. This finding aligns with previous studies [48,49], which also reported the superior performance of SVR and LSTM models in similar contexts.

Data analysis during unusual traffic conditions, such as the days before and after holidays, is presented in Table 5. The model performance results showed that the LSTM model substantially outperformed the other three models during peak periods and highly unusual congestion (7:30–9:30 for the inbound routes and 18:00–19:00 for the outbound routes).

To understand the sensitivity of the missing data, we selected the outbound routes where the traffic was heavier than on the inbound routes. Five levels of missing data were simulated (10%, 20%, 40%, 60%, and 80%) using the existing dataset. The results in Figure 16 showed that the LSTM model has the lowest MAPEs for all missing data levels for both the long and short routes. In this study, we considered only the effects of missing data on the performance of the travel time models. However, other factors, such as area type (urban/rural), number of lanes, weather, and road incidents, could potentially influence model performance as well. The inclusion of other variables, such as weather conditions (rain/on rain) and incident occurrence, might further improve model performance.

Author Contributions: The authors confirm their contributions to the paper as follows: K.J.: conceptualization, writing—review and editing, funding acquisition, project administration, and resources; C.A.: data curation, formal analysis, and investigation; S.B.: conceptualization, methodology, writing—original draft preparation; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Thailand Research Fund, grant number MRG5580224.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This study was supported by the Thailand Research Fund (Project No. MRG5580224). The authors also thank Don Mueang Tollway Public Company Limited for support in data collection.

Conflicts of Interest: The authors declare no conflict of interest. The funder had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Liu, K.; Cui, M.-Y.; Cao, P.; Wang, J.-B. Iterative Bayesian Estimation of Travel Times on Urban Arterials: Fusing Loop Detector and Probe Vehicle Data. *PLoS ONE* **2016**, *11*, e0158123. [CrossRef] [PubMed]
2. Peng, L.; Li, Z.; Wang, C.; Sarkodie-Gyan, T. Evaluation of Roadway Spatial-Temporal Travel Speed Estimation Using Mapped Low-Frequency AVL Probe Data. *Measurement* **2020**, *165*, 108150. [CrossRef]
3. Haghani, A.; Hamedi, M.; Sadabadi, K.F.; Young, S.; Tarnoff, P. Data Collection of Freeway Travel Time Ground Truth with Bluetooth Sensors. *Transp. Res. Rec.* **2010**, *2160*, 60–68. [CrossRef]
4. Sharifi, E.; Hamedi, M.; Haghani, A.; Sadrsadat, H. Analysis of Vehicle Detection Rate for Bluetooth Traffic Sensors: A Case Study in Maryland and Delaware. In Proceedings of the 18th World Congress on on Intelligent Transport Systems 2011, Orlando, FL, USA, 16–20 October 2011.
5. Chen, M.; Xia, J.; Liu, R.R. Developing a Strategy for Imputing Missing Traffic Volume Data. *J. Transp. Res. Forum* **2010**, *45*, 57–75. [CrossRef]
6. Liu, Y.; Xia, J.; Phatak, A. Evaluating the Accuracy of Bluetooth-Based Travel Time on Arterial Roads: A Case Study of Perth, Western Australia. *J. Adv. Transp.* **2020**, *2020*, 9541234. [CrossRef]
7. Jedwanna, K.; Boonsiripant, S. Evaluation of Bluetooth Detectors in Travel Time Estimation. *Sustainability* **2022**, *14*, 4591. [CrossRef]
8. Lipton, Z.C.; Kale, D.C.; Wetzel, R. Modeling Missing Data in Clinical Time Series with Rnns. *Mach. Learn. Healthc.* **2016**, *56*, 253–270.
9. Xiangxue, W.; Lunhui, X.; Kaixun, C. Data-Driven Short-Term Forecasting for Urban Road Network Traffic Based on Data Processing and LSTM-RNN. *Arab. J. Sci. Eng.* **2019**, *44*, 3043–3060. [CrossRef]
10. Chollet, F.; Allaire, J.J. *Deep Learning Mit R Und Keras: Das Praxis-Handbuch von Den Entwicklern von Keras Und RStudio*; MITP-Verlags GmbH & Co. KG: Frechen, Germany, 2018; ISBN 3958458955.
11. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support Vector Machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
12. Tan, H.; Wu, Y.; Cheng, B.; Wang, W.; Ran, B. Robust Missing Traffic Flow Imputation Considering Nonnegativity and Road Capacity. *Math. Probl. Eng.* **2014**, *2014*, 763469. [CrossRef]
13. Qu, L.; Li, L.; Zhang, Y.; Hu, J. PPCA-Based Missing Data Imputation for Traffic Flow Volume: A Systematical Approach. *IEEE Trans. Intell. Transp. Syst.* **2009**, *10*, 512–522.
14. Ni, D.; Leonard, J.D.; Guin, A.; Feng, C. Multiple Imputation Scheme for Overcoming the Missing Values and Variability Issues in ITS Data. *J. Transp. Eng.* **2005**, *131*, 931–938. [CrossRef]
15. Luo, X.; Meng, X.; Gan, W.; Chen, Y. Traffic Data Imputation Algorithm Based on Improved Low-Rank Matrix Decomposition. *J. Sens.* **2019**, *2019*, 7092713. [CrossRef]
16. Chen, J.; Shao, J. Nearest Neighbor Imputation for Survey Data. *J. Off. Stat.* **2000**, *16*, 113.
17. Beretta, L.; Santaniello, A. Nearest Neighbor Imputation Algorithms: A Critical Evaluation. *BMC Med. Inform. Decis. Mak.* **2016**, *16*, 197–208. [CrossRef]
18. Shin, D.-H.; Chung, K.; Park, R.C. Prediction of Traffic Congestion Based on LSTM through Correction of Missing Temporal and Spatial Data. *IEEE Access* **2020**, *8*, 150784–150796. [CrossRef]
19. Tang, J.; Zou, Y.; Ash, J.; Zhang, S.; Liu, F.; Wang, Y. Travel Time Estimation Using Freeway Point Detector Data Based on Evolving Fuzzy Neural Inference System. *PLoS ONE* **2016**, *11*, e0147263. [CrossRef] [PubMed]
20. Do, M.; Pueboobpaphan, R.; Miska, M.; Kuwahara, M.; van Arem, B. A Simple Data Fusion Method for Instantaneous Travel Time Estimation. In Proceedings of the 12th World Conference on Transport Research, Instituto Superior Tecnico (IST) 2010. Lisbon, Portugal, 11–15 July 2010; pp. 1–21.
21. Li, R.; Rose, G.; Sarvi, M. Evaluation of Speed-Based Travel Time Estimation Models. *J. Transp. Eng.* **2006**, *132*, 540–547. [CrossRef]
22. Xiao, Y.; Qom, S.F.; Hadi, M.; Al-Deek, H. Use of Data from Point Detectors and Automatic Vehicle Identification to Compare Instantaneous and Experienced Travel Times. *Transp. Res. Rec.* **2014**, *2470*, 95–104. [CrossRef]
23. Kwon, J.; Coifman, B.; Bickel, P. Day-to-Day Travel-Time Trends and Travel-Time Prediction from Loop-Detector Data. *Transp. Res. Rec.* **2000**, *1717*, 120–129. [CrossRef]
24. Zhang, M.; Wu, T.Q.; Kwon, E.; Sommers, K.; Habib, A. *Arterial Link Travel Time Estimation Using Loop Detector Data*; University of Iowa Public Policy Center: Iowa City, IA, USA, 1997; Volume 10, p. 2015.
25. Samuel, A.L. Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. *IBM J. Res. Dev.* **1967**, *11*, 601–617. [CrossRef]
26. Bhavsar, P.; Safro, I.; Bouaynaya, N.; Polikar, R.; Dera, D. Machine Learning in Transportation Data Analytics. In *Data Analytics for Intelligent Transportation Systems*; Elsevier: Amsterdam, The Netherlands, 2017; pp. 283–307.
27. Habtemichael, F.G.; Cetin, M. Short-Term Traffic Flow Rate Forecasting Based on Identifying Similar Traffic Patterns. *Transp. Res. Part C Emerg. Technol.* **2016**, *66*, 61–78.

28. Yu, B.; Wang, H.; Shan, W.; Yao, B. Prediction of Bus Travel Time Using Random Forests Based on near Neighbors. *Comput. Civ. Infrastruct. Eng.* **2018**, *33*, 333–350. [CrossRef]
29. Qiu, B.; Fan, W. Machine Learning Based Short-Term Travel Time Prediction: Numerical Results and Comparative Analyses. *Sustainability* **2021**, *13*, 7454. [CrossRef]
30. Vanajakshi, L.; Rilett, L.R. Support Vector Machine Technique for the Short Term Prediction of Travel Time. In Proceedings of the 2007 IEEE Intelligent Vehicles Symposium, Istanbul, Turkey, 13–15 June 2007; IEEE: New York, NY, USA, 2007; pp. 600–605.
31. Hou, Y.; Edara, P. Network Scale Travel Time Prediction Using Deep Learning. *Transp. Res. Rec.* **2018**, *2672*, 115–123. [CrossRef]
32. Islek, I.; Oguducu, S.G. Use of LSTM for Short-Term and Long-Term Travel Time Prediction. In *Proceedings of the CEUR Workshop Proceedings; the Creative Commons License Attribution 4.0 International*; CEUR-WS: Aachen, Germany, 2019; Volume 2482.
33. Liu, Y.; Wang, Y.; Yang, X.; Zhang, L. Short-Term Travel Time Prediction by Deep Learning: A Comparison of Different LSTM-DNN Models. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; IEEE: New York, NY, USA, 2017; pp. 1–8.
34. Bai, C.; Peng, Z.-R.; Lu, Q.-C.; Sun, J. Dynamic Bus Travel Time Prediction Models on Road with Multiple Bus Routes. *Comput. Intell. Neurosci.* **2015**, *2015*, 432389. [CrossRef]
35. Leys, C.; Ley, C.; Klein, O.; Bernard, P.; Licata, L. Detecting Outliers: Do Not Use Standard Deviation around the Mean, Use Absolute Deviation around the Median. *J. Exp. Soc. Psychol.* **2013**, *49*, 764–766. [CrossRef]
36. Orr, J.M.; Sackett, P.R.; Dubois, C.L.Z. Outlier Detection and Treatment in I/O Psychology: A Survey of Researcher Beliefs and an Empirical Illustration. *Pers. Psychol.* **1991**, *44*, 473–486. [CrossRef]
37. 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]
38. Cortes, C.; Vapnik, V. Support-Vector Networks. *Mach. Learn.* **1995**, *20*, 273–297.
39. Gunn, S.R. Support Vector Machines for Classification and Regression. *ISIS Tech. Rep.* **1998**, *14*, 5–16.
40. Wu, C.-H.; Ho, J.-M.; Lee, D.-T. Travel-Time Prediction with Support Vector Regression. *IEEE Trans. Intell. Transp. Syst.* **2004**, *5*, 276–281. [CrossRef]
41. Liu, Y.; Ji, Y.; Chen, K.; Qi, X. Support Vector Regression for Bus Travel Time Prediction Using Wavelet Transform. *J. Harbin Inst. Technol. (New Ser.)* **2019**, *26*, 26–34.
42. Graves, A. Long Short-Term Memory. In *Supervised Sequence Labelling with Recurrent Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 37–45.
43. Abbas, Z.; Al-Shishtawy, A.; Girdzijauskas, S.; Vlassov, V. Short-Term Traffic Prediction Using Long Short-Term Memory Neural Networks. In Proceedings of the 2018 IEEE International Congress on Big Data (BigData Congress), Seattle, WA, USA, 10–13 December 2018; IEEE: New York, NY, USA, 2018; pp. 57–65.
44. Fu, R.; Zhang, Z.; Li, L. Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction. In Proceedings of the 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC), Wuhan, China, 11–13 November 2016; IEEE: New York, NY, USA, 2016; pp. 324–328.
45. Lewis, C.D. *Industrial and Business Forecasting Methods: A Practical Guide to Exponential Smoothing and Curve Fitting*; Butterworth-Heinemann: Oxford, UK, 1982; ISBN 0408005599.
46. Thissen, U.; Van Brakel, R.; De Weijer, A.P.; Melssen, W.J.; Buydens, L.M.C. Using Support Vector Machines for Time Series Prediction. *Chemom. Intell. Lab. Syst.* **2003**, *69*, 35–49. [CrossRef]
47. Cui, Z.; Ke, R.; Pu, Z.; Wang, Y. Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Forecasting Network-Wide Traffic State with Missing Values. *Transp. Res. Part C Emerg. Technol.* **2020**, *118*, 102674. [CrossRef]
48. Zhang, H.; Wu, H.; Sun, W.; Zheng, B. Deeptravel: A Neural Network Based Travel Time Estimation Model with Auxiliary Supervision. *arXiv Prepr.* **2018**, arXiv:1802.02147.
49. Chen, D.; Yan, X.; Li, S.; Wang, L.; Liu, X. Long Short-Term Memory Neural Network for Travel Time Prediction of Expressways Using Toll Station Data. In Proceedings of the CICTP 2020, Xi'an, China, 14–16 August 2020; pp. 73–85.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.