Research Article

Mapping Bus and Stream Travel Time Using Machine Learning Approaches

Gopikrishnan Nair 1, B. Anil Kumar 1, and Lelitha Vanajaskshi 2

1Department of Civil and Environmental Engineering, Indian Institute of Technology Patna, Patna-801103, India
2Department of Civil Engineering, Indian Institute of Technology Madras, Chennai-600036, India

Correspondence should be addressed to Lelitha Vanajaskshi; lelitha@iitm.ac.in

Received 17 May 2021; Revised 17 March 2022; Accepted 18 March 2022; Published 25 April 2022

Copyright © 2022 Gopikrishnan Nair et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Collection of travel time data has always been a strenuous task, especially on Indian roads, due to the highly mixed traffic conditions and the absence of rigid driving characteristics. Travel time data collection methods such as on-board GPS devices and Wi-Fi scanners have their own feasibility issues. The GPS devices cannot be installed on all private vehicles, and Wi-Fi scanners cannot be set up at each and every road corridor. However, many transit agencies of Indian cities are installing GPS units in their public transport buses, making them a rich source of data. These buses travel all over the network and record the travel time and location information of vehicles at a certain interval, which can become the best and reliable source of traffic data under such conditions. The only limitation with this data is that the buses differ greatly from the rest of the vehicles in terms of vehicle characteristics, freedom of driving, effect of bus-stops, etc. So, a viable solution is to model and link the stream and bus travel times. Based on this, the present study proposes a method to map bus travel times with stream travel times using machine learning techniques, namely, Gradient Boosting Method and Support Vector Machines. Preliminary analysis showed that a nonlinear relationship exists between the travel times, and other factors like peak/off-peak hours, day of the week, etc. Results show that the proposed methods can efficiently map bus travel times with stream travel times with an average MAPE of 20% and perform better than existing approaches such as linear regression and artificial neural networks.

1. Introduction

The challenges related to traffic faced by urban areas in both developed and developing countries are infrastructure deficiency, congestion, accidents, and environmental and health damages due to pollution. Though all of these are important, the problem of traffic congestion is the most visible and affecting the mobility of large number of commuters directly on a day-to-day basis, leading to hour-long queues and longer commutation times. For example, recent surveys on average daily travel times reveal that commuters spent an extra two and half hours on the roads [1] due to increased vehicles and congestion. In this regard, developing Smart Mobility Solutions can be viable in reducing and mitigating the negative effects of increased transportation demand on existing infrastructure without resorting to building new roadways, widening existing roads, etc. Providing information on travel times to commuters in advance is one such solution [2]. It helps individual riders make an accurate approximation for the duration of travel and also pick the best route for their travel. From a logistics point of view, this will help in improving the standard of customer service by making on-time deliveries, thus reducing the operational costs. From a traffic engineer’s point of view, it helps improve the efficiency and maximize utilization of available routes in the road network. Thus, prediction results can help the city traffic management analyze and identify the flaws in the infrastructure and come up with alternative solutions. If there is an accurate travel time prediction available across all routes to the user, it will have a huge positive impact on traffic congestion across the global road network. In order to achieve such benefits, it is essential to have reliable and accurate source of travel time data.
Travel time data can be collected through a wide variety of methods [3, 4]. However, they all involve challenges such as infrastructure requirement, real-time communication, and continuous power supply requirements. Since travel time is spatial in nature, the vehicles need to be continuously tracked or reidentified at one or more locations for its measurement. At an individual level, one can collect the travel time by recording his/her time using a stopwatch to travel a particular distance. More general methods, which do not involve the individual traveler to collect travel times, are license plate matching, video cameras, Wi-Fi, Bluetooth scanners, stream-based techniques such as test vehicles, Automatic Vehicle Identifiers (RFID), etc. In the Indian context, lack of lane discipline and heterogeneity in the vehicle classes makes it challenging to reidentify vehicles using traffic sensors such as number plate recognition and video image processing techniques. Under such conditions, tracking the vehicles and measuring travel time using Global Positioning System (GPS) and advanced sensors such as Wi-Fi and Bluetooth scanners are becoming more popular due to its (i) low operating cost and affordability and (ii) ability to work independent of the traffic condition. One of the main advantages of these techniques is that the travel time can be directly obtained, and they are mostly nonintrusive [5, 6].

In recent times, researchers started exploring the use of Bluetooth Media Access Control (MAC) address matching to estimate travel times [7–10]. However, due to increased security on mobile devices and in-car navigation systems, their penetration rate appears to be declining [11, 12]. On the other hand, the usage of Wi-Fi has been consistently increasing over the last several years and has motivated researchers to develop Wi-Fi MAC scanners to estimate travel times [12]. However, Indian cities currently do not have established infrastructure to obtain such data continuously for long periods at network level, and there is a need to look into alternative option. GPS fitted public transport vehicles are part of every smart cities, and they are one guaranteed and scalable solution available now. The use of transit vehicles as probes for travel time data collection does not require any additional infrastructure beyond what is already used by the bus system, without having any privacy concerns. Although buses can be used as probe vehicles, they cannot be treated as regular probe vehicles, as bus travel time characteristics may be different from the other vehicles due to their frequent stopping at bus-stops and due to the difference in their vehicular characteristics. Thus, if one has to estimate stream travel time using the bus GPS data, a suitable methodology to map both bus and stream travel time data needs to be developed and is the main aim of this study.

A few studies examined the possibility of using regular probe vehicles and buses as probes to estimate stream travel time. Using loop detector data [13–18], AVI data [19], and probe vehicles that are equipped with GPS [20–23] to estimate travel time in freeways and corridors under homogeneous traffic conditions was reported in literature. Elango and Dailey [24], Cathey and Dailey [25], Bertini and Tantiyanugulchait [26], and Uno et al. [27] explored the use of buses as probe vehicles to measure speeds and to analyze travel time variability. Chakroverty and Kikuchi [28] examined the use of transit vehicles as probe vehicles to estimate stream travel time on urban corridors using regression models. Forouzandeh et al. [29] used Holt-Winters analysis to predict stream travel time using buses as probes. Kumar et al. [30] developed a model-based approach to predict stream travel times from bus travel time data using Kalman filtering technique. Sakhare and Vanajakshi [11] developed regression and artificial neural network models to estimate the stream travel time using bus travel time data obtained from GPS devices. The methodologies followed in the existing literature can be classified into two broad categories: (i) statistical estimation techniques such as regression analysis and cross-correlation and (ii) machine learning approaches such as Artificial Neural Networks (ANNs). However, these methods may not be able to capture the variations in travel time when the variability is high. In order to capture the nuances of travel time variation, both due to the day of the week and due to the time of the day, this study proposes the use of Gradient Boosting Method (GBM) and Support Vector Machines (SVMs). This is mainly because GBM is reported to be one of the best among many existing machine learning approaches, because of its ability (i) to fit complex nonlinear relationship between variables, (ii) to deal with multicollinearity between variables and to avoid overfitting [31, 32], and (iii) to handle different types of predictor variables and interpretability, unlike other machine learning methods. Alongside this, the present study also uses Support Vector Machine (SVM) that is also reported to have an ability to build complex relationships between variables when the variability in data is very high [33]. Hence, these approaches warrant a more capable model that is effectively able to reflect the actual travel time variation on the roads.

Overall, from the existing literature, it can be observed that only limited studies explored the use of probe vehicles as possible source of traffic data [13–18], and very few utilized the use of transit vehicles as probe vehicle [11, 28–30, 34], and none of them considered the effect of intraday variations in travel time and effect of the day of the week in the modelling. For example, both transit probe vehicles and stream may behave similarly during peak hours and different during off-peak hours. A similar case can be expected during weekdays and weekends. Thus, the main objectives of the present study are as follows:

(1) Analyzing the travel time of transit buses using GPS units and stream using Wi-Fi MAC matching technique to identify patterns in the data.
(2) Identifying the model features/independent variables based on the patterns in travel time.
(3) Developing a methodology that can efficiently map the bus travel time to the average stream travel time.
(4) Corroborating the developed estimation method and comparing the performance with reported studies.
2. Methodology

In order to estimate the stream travel time using the travel time data collected from transit vehicles, the present study followed a three-step approach as shown in Figure 1. As part of the first step, the study stretch was identified, and required data were collected using GPS and Wi-Fi scanners. Then, the collected data were analyzed to check the suitability of using transit probe data to estimate stream travel time and identified the independent variables to be included in the model by observing the inherent patterns in travel time data. In the third step, the methods based on GBM and SVM were developed to estimate stream travel time, and the performance was evaluated with the filed data and compared the performance with existing approaches [11]. Each of the steps presented in Figure 1 are detailed in the following sections.

3. Study Area and Data Collection

The data used in this study were collected using GPS devices fitted inside buses and Wi-Fi scanners placed alongside of roads. Four different sections with heterogeneous traffic and varying land use characteristics were considered for this study in the city of Chennai, India, and corresponding details are presented in Table 1. The route details are shown in Figure 2. These sections with varying geometric conditions, land use and traffic volume, were selected to demonstrate the robustness of the models under different conditions.

Wi-Fi and Bluetooth monitoring stations were used to collect stream travel time, and the travel times of transit probes were collected using permanently fitted GPS units in Metropolitan Transport Corporation (MTC) buses that are plying in the selected sections.

Wi-Fi MAC data were collected using five monitoring stations at Vijaya Nagar, TCS, Little Mount, Indira Nagar, and CSIR as shown in Figure 2. Wi-Fi scanners developed in-house [12] were used to collect Wi-Fi MAC data from vehicles. It performs dual scanning, signal strength filtering, and MAC hashing to maintain privacy. In general, each Wi-Fi enabled device will have a unique identifier, i.e., MAC ID, and when the scanner communicates with a Wi-Fi enabled device (smartphones/tablets/media devices) in the stream, it responds to the inquiry scan with its unique MAC ID, signal strength, and timestamp. Real time communication of this data was made possible through General Packet Radio Service (GPRS), and the data received from each device were stored in a server using Structured Query Language (SQL) database. The stored data were further used to estimate travel times using MAC address matching technique [9, 12, 35].

Travel time data of transit vehicles were collected using permanently fitted GPS units in MTC buses in the city of Chennai. Two MTC bus routes running in the selected corridors, namely, M1 and 19B, were considered. The collected GPS data included the latitude and longitude information at fixed time intervals (10 seconds), time stamp corresponding to each entry and ID of the GPS units. The raw GPS data were then processed to extract section travel times using in-house methodology developed by Koppineni et al. [36]. Once travel times are computed, outliers were identified by applying thresholds and Inter Quartile Range based on speed. Thresholds were set considering the maximum speed of the road as 80 km/h and the minimum speed of 5 km/h. Following this, IQR based outlier removal was done, and the resulting data was used for the modelling. A total of 1141 bus trips of data were collected during the study period in the selected corridors. In order to map both bus and stream travel times, data collected from both sources were matched based on their entry time into the study area. For a particular bus entering into the corridor, it was assumed that all stream vehicles that enter the study area within a range of 5 minutes (±2.5 minutes). The median of all such stream travel times was calculated and matched to the bus travel time to form the dataset required for the modelling. Out of the collected data, 80% of the data were used to train the model, and the remaining data were used to test the model.

4. Preliminary Analysis

To start, the acquired data were analyzed to check the suitability of using transit probe data to estimate stream travel time. For this, cumulative frequency diagrams (CFD) of stream and transit probe travel times for a period of one month were plotted for comparison purposes as shown in Figure 3.

From Figure 3, it can be observed that (i) pattern of bus travel times and stream travel times are similar, and (ii) bus travel times are always higher than stream travel times with an average travel time difference of 60 s for TCS to VN section and 130 s for VN to TCS section. This may be due to the frequent stopping of buses at bus-stops, freedom of driving being constrained for bus due to bigger size, lane usage, etc. For an ideal comparison of travel times, dwell times and associated acceleration/deceleration times of the buses at each stop must be removed. However, due to the low polling rate of on-board GPS devices, the whole trajectory along the length was considered, and the number of bus-stops has been included as a feature in the model development.

Further, to inspect whether the similarity between bus and stream travel times, as observed in Figure 3, is statistically significant, a correlation analysis was carried out by calculating a measure of correlation, Pearson Correlation coefficient (r) as

\[ r = \frac{n\sum xy - \sum x \sum y}{\sqrt{n\sum x^2 - (\sum x)^2} \sqrt{n\sum y^2 - (\sum y)^2}}, \]

where \( n \) is the number of observations, and \( x \) and \( y \) are the stream and bus travel times.

In general, Pearson Coefficient values lie in the range of [–1, 1]. A Pearson coefficient value of –1 implies a perfect negative linear relationship, while a value of +1 implies a perfect positive linear relationship, and a value of 0 is an indication of very weak or no linear relationship at all. Following literature, correlation values less than +0.8 and greater than –0.8 are considered to be insignificant [37, 38].
in this study. The correlation analysis was conducted on two selected routes, and a sample result is shown in Figure 4. From Figure 4, it can be seen that the correlation values are ranging from 0.1 to 0.4 for Vijaya Nagar to TCS section and 0.6 to 0.8 for TCS to Vijaya Nagar section. Overall, from Figure 4, it can be seen that the results are inconclusive of a strong linear relationship in this section. As it failed to show any significant evidence of linear relationship, the present study took up an approach of building nonlinear models using GBM and SVM to establish a robust mapping between bus and stream travel times.

Different times of the day/day of the week may have different levels of congestion on the section, and this may have different effect on the movement of buses compared to the other stream vehicles. Therefore, the acquired data were further analyzed for patterns. Traffic on any given section is expected to follow temporal patterns with time of the day and day of the week. Identification of these patterns in data will help in identifying the features to be included in the modelling part. In order to visualize the effect of time of the day and day of the week on travel time, its variation throughout the day was plotted as shown in Figure 5. From Figure 5(a), it can be observed that there are two visible peaks: one in the morning and one in the evening. Based on this, the following groupings were made:

(i) Two peak periods: mild morning peak (AM peak) from 9 AM to 1 PM and pronounced clear evening peak (PM peak), from 4 PM to 11 PM.
(ii) Two off-peak periods: morning off-peak (AM off-peak), from 5 AM to 9 AM and afternoon off-peak (PM off-peak) from 1 PM to 4 PM.
(iii) Late night hours, from 11 PM to 5 AM.

Along with this, from Figure 5(b), it can be observed that the travel times are varying for each day of the week. Figure 5, coupled with the fact that the vehicular characteristics of buses are different from those of other vehicles, led to the inclusion of more independent variables in the modelling namely, peak/off-peak distinction along with day of the week. Therefore, the input groups used are as follows:

(i) Number of bus-stops in a section, as the dwell time has not been accounted in the raw data due to high polling rate.
(ii) Time of the day on the basis of hourly analysis.
(iii) Day of the week, as analysis showed a distinct difference between each day of the week.

5. Model Development

In order to build a robust nonlinear relationship between variables to estimate stream travel time, the present study proposes to use Gradient Boosting Method (GBM) that is reported to be one of the best among many existing machine learning approaches along with Support Vector Machine (SVM) that has been reported to have ability to build complex relationships between variables. A brief discussion on GBM and SVM is presented below along with involved parameters and details of implementation. Out of the collected data, 80% of the data were used to train the model, and the remaining data were used to test the model.
Figure 2: The study corridors and their corresponding Wi-Fi stations (source: Google Maps). (a) VN to TCS/TCS to VN. (b) Little mount to Indira Nagar. (c) Indira Nagar to CSIR.

Figure 3: Sample CFD of bus and stream travel times. (a) TCS to Vijaya Nagar. (b) Vijaya Nagar to TCS.
Gradient Boosting Methods are widely used for their ability to learn complex relations between data with high variability [31, 32]. The present study makes use of an implementation of Gradient Boosting for regression and thereby predicts stream travel times.

Boosting refers to a technique that is used to transform what were originally poor learning models into far better models. The technique makes use of multiple algorithms sequentially so as to create a model that is a good learner and predictor. The gradient boosting algorithm makes use of decision trees.

Figure 4: Correlation between bus and stream travel times.

Figure 5: Variations in stream travel time with respect to time of the day and day of the week (TCS-VN link). (a) Time of the day. (b) Day of the week.
as its base and builds upon it to become a better model. The basic principle behind it is that it uses multiple decision trees, which are trained in a sequential manner. Boosting fits additional models that minimize a certain loss function averaged over the training data, such as a root mean squared-error and absolute error. The loss function measures how inaccurate the prediction is, and the weights will be adjusted based on how much the predicted value deviates from the true value. The first tree in the sequence predicts the observations by giving equal weights to all observations. The next tree will change the weights for the observation such that it can be used to predict the previous errors. The process continues until the specified number of trees is used for the prediction. The most optimum results will be achieved when the error is minimized. In terms of regression problem, the boosting method is a form of "functional gradient decent." It is an optimization technique that minimizes a certain loss function by adding a base model at each step that best reduces the loss function.

The GBM has numerous hyperparameters and can be classified into two categories: (i) general parameters that define the algorithm as a whole and (ii) tree specific parameters that define the characteristics of each tree in the model. The general parameters include learning rate ($j$) and the number of trees ($M$) in the algorithm that can define the algorithm as a whole. The tree specific parameters include maximum depth of the tree, minimum number of observations per node to perform a split, and minimum number of observations required per leaf. These parameters can be tuned to achieve optimum performance [39]. For example, by increasing the number of iterations, the model becomes complex, and minor fluctuations in the data may be overstated/overfitted and can lead to poor performance on testing data. Therefore, it is necessary to determine optimum number of trees to minimize the risks that are associated with the robust estimation. In the present study, for optimizing and fine-tuning the model, a grid search cross validation was implemented. It loops through a given range of parameters to find the optimum combination and finally fits the model to the training data. This final model was used to test on the new data from the study corridors to estimate corresponding stream travel times. Table 2 shows the list of parameters used to train the GBM model, and the pseudocode used for implementing the gradient boosting method is shown in Algorithm 1.

5.2. Support Vector Machines. Support Vector Machines (SVMs) are a classification-based learning algorithm and can be used for modelling nonlinear relationships too. The basic idea of support vector regression (SVR) is to map the data into high-dimensional feature space via nonlinear mapping and perform linear regression in this space with the use of kernel functions. Kernels are transformation equations, which can transform the data into a higher dimensional space. In this higher dimension, a hyperplane can be constructed between the two sets of data points such that the margin of separation between the two is maximized. In simple words, such linear regression in high-dimension space is equivalent to nonlinear regression in the low-dimension input space to better understand complex relationships between dependent and independent variables. The parameters that can be used to fine-tune SVMs include (i) Regularization Parameter ($C$), (ii) Coefficient of the support vectors ($\text{coef}$), (iii) Epsilon ($\varepsilon$), and (iv) Gamma ($\gamma$). The regularization parameter is used to balance the penalty that is imposed on the points with errors in classifications. The gamma parameter controls the influence of the support vectors (i.e., the data points closest to the plane). The epsilon parameter refers to the margin of tolerance such that a penalty is not imposed on the data points.

As the data used in the present study does not justify a linear relationship between the travel times, an SVM with the Radial Basis Function (RBF) type kernel was used. In order to identify optimum parameters, a grid search cross validation was implemented. For building the model, 80% of the data from each corridor was used to train the model and the rest 20% for the testing. The independent variables used were: (i) the bus travel time as a continuous variable, (ii) the number of bus-stops as a continuous variable, (iii) the peak hours as a categorical variable, and (iv) the day of the week as a categorical variable and stream travel time as dependent variable. Table 3 shows the list of parameters used to train the SVM model, and the pseudocode used for implementing the SVM is presented in Algorithm 2.

6. Results

The results obtained from the implementation of the GBM and SVM algorithms presented in previous section are discussed here. The error metrics Mean Absolute Percentage Error (MAPE) and the Root Mean Squared Error (RMSE) were used to quantify the accuracy of the proposed method. Further, the performance of the proposed method was compared with two existing approaches, linear regression (LR) method and artificial neural networks [11].

6.1. Performance Evaluation. To start, the performance of the proposed methods was corroborated with the field data, and corresponding results are shown in Figure 6 for a sample stretch, TCS to Vijaya Nagar corridor of length 1.6 km that has four bus-stops.

From Figure 6, it can be observed that the estimations made by proposed methods are closer to actual travel time and can capture the variations. The RMSE for this case was 1.5 minutes. The results obtained for all the sections under consideration are shown in Table 4 and 5 in terms of MAPE and RMSE, respectively. It can be seen that GBM performed better than SVM in all the cases with an average MAPE of 4% and RMSE of 20 seconds. In order to check the robustness of the proposed approaches, obtained results were corroborated with the filed data and analyzed for peak and off-peak hours of the day and various days of the week separately. Figure 7 shows the errors obtained from the GBM approach and SVM approach for one-hour interval along TCS-VN and VN-TCS links. From the figure, it can be observed that GBM can perform better than SVM during both off-peak
and peak periods of the day. In the next level, the performance of the proposed methods was evaluated for each day of the week as shown in Figure 8.

From Figure 8, it can be observed that both methods can capture the intraweek travel time variations consistently. The ability of the model to reflect the trend of the actual travel times is promising. It can also be observed from the figure that GBM can perform better than SVM.

6.2. Performance Comparison. Next, the performance of the proposed method was compared with existing approaches, namely, linear regression (LR) method and artificial neural networks that were developed under similar traffic conditions [11]. To maintain parity, both these approaches were trained with similar set of independent and dependent variables as that of GBM and SVM. For constructing ANN model, a multilayer feedforward network with the

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Range of values</th>
<th>Calibrated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No. of trees</td>
<td></td>
<td>2600</td>
</tr>
<tr>
<td>2</td>
<td>Maximum depth</td>
<td>Positive integers</td>
<td>8</td>
</tr>
<tr>
<td>3</td>
<td>Maximum samples split</td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>Maximum samples leaf</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Learning rate</td>
<td>(0, 1)</td>
<td>0.01</td>
</tr>
<tr>
<td>6</td>
<td>Subsample size</td>
<td>(0, 1)</td>
<td>0.4</td>
</tr>
<tr>
<td>7</td>
<td>Maximum number of features</td>
<td>&quot;auto,&quot; &quot;sqrt,&quot; &quot;log2,&quot; int values, float values</td>
<td>&quot;sqrt&quot;</td>
</tr>
</tbody>
</table>

**Algorithm 1:** Pseudo Code for implementation of GBM

1. Model Parameters: Number of trees (M), learning rate (J), variable interactions (C);
2. Input: [bus travel time, time of the day, day of the week, no. of bus-stops];
3. Output: [stream travel time];
4. Sequentially make the "n" number of trees:
   (a) Build the initial decision tree to predict the observations for all the \([x_1, x_2, \ldots, x_n, y]\) in the training set.
   (b) Calculate loss at each step is calculated using the loss function, \(L_f\).
   (c) Calculate the negative gradient of the loss using \(g(x) = -\frac{\partial L(y, f(x))}{\partial f(x)}\).
   (d) Predict the negative gradients of the loss functions using the set \([x_1, x_2, \ldots, x_n, g(x)]\) as the training data in subsequent decision trees.
   (e) Calculate the multiplier (\(\beta\)) using the gradient value.
   (f) Build a final model that is an additive sequence of all the decision trees, as \(F(x) = f(x) + \beta g(x)\).
5. For every \([x_1, x_2, \ldots, x_n, y]\) in the test dataset: Use the final model to predict on the testing data, \(y_{pred}\).

<table>
<thead>
<tr>
<th>No.</th>
<th>Parameter</th>
<th>Range of values</th>
<th>Calibrated value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regularization parameter (C)</td>
<td>Float values</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Kernel</td>
<td>&quot;linear,&quot; &quot;poly,&quot; &quot;rbf,&quot; &quot;sigmoid,&quot; &quot;precomputed&quot;</td>
<td>rbf</td>
</tr>
<tr>
<td>3</td>
<td>Gamma</td>
<td>&quot;scale,&quot; &quot;auto,&quot; float values</td>
<td>Scale</td>
</tr>
<tr>
<td>4</td>
<td>Coefficient</td>
<td>Float values</td>
<td>0</td>
</tr>
</tbody>
</table>

**Algorithm 2:** Pseudo Code for implementation of SVM.

1. Model Parameters: C (Regularization/Penalty parameter), y (width parameter/kernel coefficient), Epsilon (\(\epsilon\)); Kernel-type: RBF;
2. Input: [bus travel time, time of the day, day of the week, no. of bus-stops];
3. Output: [stream travel time];
4. For every \([x_1, x_2, \ldots, x_n, y]\) in the training dataset:
   (a) Transform the points into a kernel space using the kernel equations as defined,
   (b) Calculate the distance from the points to a hyperplane in the n-dimensional space such that \(|y - f(x)| < \epsilon\).
   (c) Apply a penalty for points that do not satisfy condition,
   (d) Optimize/minimize this distance to find the ideal hyperplane.
5. For every \([x_1, x_2, \ldots, x_n, y]\) in the test dataset:
   Find closest value to the corresponding hyperplane for given set of predictors, \(y_{pred}\).
Levenberg-Marquardt backpropagation algorithm was used for training, and a hyperbolic tangent sigmoid function was used as the transfer function for both the hidden layer and the output layer. The linear regression model was trained using the Ordinary Least Squares (OLS) method. Figure 9 shows the errors obtained from linear regression and artificial neural networks along with proposed gradient boosting and support vector machine approaches.

From Figure 9, it can be observed that the proposed approaches can perform better than the existing approaches. The results obtained from proposed and existing approaches are also presented at an aggregated level in Table 4 in terms of MAPE, and in Table 5 in terms of RMSE.

From Table 4, it can be observed that the GBM method has an advantage (in terms of MAPE) up to 6%, 12%, 27%, and 7% over linear regression approach and 3%, 5%, 12%, and 3% over ANN for selected sections. SVM also performed better than the earlier methods with 4%, 11%, 15%, and 5% over linear regression approach and 3%, 5%, 12%, and 3% over ANN for selected sections. The values in bold indicate the lowest error among the four proposed methods for each of the study corridor. It is intended to show the best performing model. For example, the value 18.54 in bold represents that GBM method performs better than SVM, LR, and ANN approaches in TCS to VN link.

From Table 5, it can be observed that GBM method has advantage (in terms of RMSE) up to 35 seconds over linear regression approach along the TCS to VN section., 13 seconds of advantage along VN to TCS section, 60 seconds of advantage along LM to Indira Nagar section, and 36 seconds of advantage along Indira Nagar to CSIR section. A similar observation can be seen for the SVM as well, reinforcing the superiority of the proposed methods. Overall, from Tables 4 and 5, it can be observed that the proposed methods can perform better than existing
approaches and within proposed approaches, GBM can perform better than SVM in most of the cases.

7. Summary and Conclusions

Travel time, a fundamental measure in transportation, can be defined as the time taken to traverse a route between any two points of interest. It is essential to have reliable and accurate source of travel time data, particularly in the context of developing smart mobility solutions. However, it is difficult to collect travel time data, as it is a spatial variable. GPS devices that are installed on vehicles are one source of travel time. Still, due to privacy issues, public transport buses are the only potential source of collecting GPS based travel time data at network level. Characteristics of bus travel times differ greatly from the rest of the vehicles in terms of vehicle

Figure 7: Performance evaluation of proposed methods with respect to time of the day. (a) TCS to Vijaya Nagar. (b) Vijaya Nagar to TCS.

Figure 8: Performance evaluation of proposed methods with respect to day of the week.
characteristics, speeding behavior during various times of the day and day of the week, frequent stoppage at bus-stops, associated acceleration and deceleration characteristics, etc. Therefore, there is a need to build a robust estimation scheme that can map bus travel times with stream travel times. In order to validate the results, Wi-Fi scanners that capture all-modes travel times in a traffic stream are used. Data analysis showed that there exists a nonlinear relationship between bus and stream travel times, and hence, the present study mapped bus travel times with stream travel times using Gradient Boosting Method and Support Vector Machines, which are reported to be more appropriate when the underlying relations are complex and nonlinear. Next, these models were trained considering (i) the bus travel time, (ii) the number of bus-stops, (iii) traffic condition within a day (peak/off-peak), and (iv) day of the week, as independent variables and stream travel time as dependent variable. Results showed that the estimations made by the proposed methods are closer to actual travel time and can capture the variations. It was also observed that the proposed approaches can perform better than existing approaches such as linear regression and ANN, showing the efficacy of the proposed method.

The proposed methodology is a general one and can be used if GPS data and Wi-Fi-MAC data are available. The main challenge in using this approach is the requirement of a sufficiently large data set (say a week) to build models. The model parameters may have to be calibrated for any new location. However, the methodology proposed is generic and transferable. The accuracy of the proposed methods may be improved further by explicitly incorporating data related to traffic volume, weather conditions, and composition of traffic in the selected corridors. Use of more sophisticated methods such as deep learning is another future research direction.

Data Availability

Data are available upon request.

Conflicts of Interest

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

Acknowledgments

The authors acknowledge the support for this study as a part of the SPARC project funded by the Ministry of Human Resource Development, Government of India, through Project Number CIE/18-19/289/SPAR/LELI.

References


