Contents lists available at **SCCE**

Journal of Soft Computing in Civil Engineering

Journal homepage: www.jsoftcivil.com

Mode-Wise Corridor Level Travel-Time Estimation Using Machine Learning Models

R.B. Sharmila[1*], N.R. Velaga[2]

1. Research Scholar, Transportation Systems Engineering, Civil Engineering Department, Indian Institute of Technology Bombay, Mumbai, India
2. Associate Professor, Transportation Systems Engineering, Civil Engineering Department, Indian Institute of Technology Bombay, Mumbai, India

*Corresponding author: *sharmibab@gmail.com*

https://doi.org/10.22115/SCCE.2020.215679.1164

ARTICLE INFO

Article history:
Received: 13 January 2020
Revised: 04 March 2020
Accepted: 08 March 2020

Keywords:
Machine learning;
Travel-time;
Support vector machines;
Artificial neural networks.

ABSTRACT

This research is oriented towards exploring mode-wise corridor level travel-time estimation using Machine learning techniques such as Artificial Neural Network (ANN) and Support Vector Machine (SVM). Authors have considered buses (equipped with in-vehicle GPS) as the probe vehicles and attempted to calculate the travel-time of other modes such as cars along a stretch of arterial roads. The proposed study considers various influential factors that affect travel time such as road geometry, traffic parameters, location information from the GPS receiver and other spatiotemporal parameters that affect the travel-time. The study used a segment modeling method for segregating the data based on identified bus stop locations. A k-fold cross-validation technique was used for determining the optimum model parameters to be used in the ANN and SVM models. The developed models were tested on a study corridor of 59.48 km stretch in Mumbai, India. The data for this study were collected for a period of five days (Monday-Friday) during the morning peak period (from 8.00 am to 11.00 am). Evaluation scores such as MAPE (mean absolute percentage error), MAD (mean absolute deviation) and RMSE (root mean square error) were used for testing the performance of the models. The MAPE values for ANN and SVM models are 11.65 and 10.78 respectively. The developed model is further statistically validated using the Kolmogorov-Smirnov test. The results obtained from these tests proved that the proposed model is statistically valid.

How to cite this article: Sharmila RB, Velaga NR. Mode-wise corridor level travel-time estimation using machine learning models. J Soft Comput Civ Eng 2020;4(1):72–97. https://doi.org/10.22115/SCCE.2020.215679.1164.

1. Introduction

In India, around 26% of commuters are traveling for about 90 minutes every day. Although the average commuting length is 29 minutes in India, a large group of commuters travels more than an hour to and from the place of work [1]. Thus, congestion becomes a persistent problem in the Indian traffic scenario due to the heterogeneous traffic, lack of discipline, etc. Therefore, extracting travel-time information for individual modes separately becomes a tedious task. In this study, authors have attempted to use public transport (i.e., buses) as probe vehicles to estimate the travel time of the test mode (i.e., car) for an arterial corridor.

The current study chose public transit (bus transit) as the probe mode; because there will be no prevalence of privacy issues in using the public buses. The probe vehicles can be fitted with GPS receivers and can be easily tracked without much interference. Also, by using the bus as a probe mode, it is possible to acquire a large sample of data as the buses may take multiple runs on the arterial corridor in a day.

It is a well-established fact that the majority of the developed nations like the United States and European countries prefer private transport over public transport for better accessibility [2–6]. However, the scenario is different in developing nations like India, China, etc., where the majority of people rely on public transport than private vehicles [7]. In this study, authors have considered public transit (bus) as the ideal choice for probe vehicles. As public transport follow a regular pattern such as predefined route and schedule, it is easier to track them and use them as probe vehicles [8]. Thus, the bus is chosen as a viable probe mode for model development. However, there are some typical shortcomings in public transport travel-time calculations.

Issues involving public transit travel-time estimation are:

1) The public transport buses vary in travel speeds and transfer times along different routes [9].

2) In certain cases of missing information on the schedule, the average travel speeds are assumed for the whole route [10–12].

3) In most cases, the transfer waiting times are ignored or assumed as a constant[2,13].

4) Very few studies have incorporated scheduled arrival or departure times for travel time analysis [9].

Some of the existing studies have used buses as the probe vehicles and calculated the travel-time for the other modes. They have conducted a correlation analysis between the two modes (bus and other modes like a car, two-wheeler, auto, etc) and developed linear regression models. The idea of developing non-linear models using machine learning algorithms was limited for mode-wise travel-time estimation. Moreover, very limited studies have included dwell time in their models. Existing research has reported few simulation-based studies too and real-time based field scenario studies were very limited. There are many factors affecting travel-time; some are categorized as traffic factors (distance, speed, travel-time, intersection length), road geometry related factors (road width, gradients) and vehicle characteristics (bus stop halt time, passenger count). It is very difficult to develop a model that includes all these factors together and very limited research is identified where all these factors are considered together.

The above issues are addressed in this study. This study considers the non-linear relationship between the two modes (Car and Bus) using ANN and SVM. Transit buses as probes offer a number of advantages: (1) covering a large portion of urban networks; (2) GPS are installed by transit operators making the model cost-effective; and (3) privacy concerns can be eliminated by installing GPS devices in public transit systems (buses). However, there are certain drawbacks in using buses; for instance, the characteristics of the buses differ slightly in comparison to the other vehicle characteristics. The difference between the bus travel-time and the average travel time of the stream is a random variable and modeling them for other modes is a great challenge. Despite these randomness and biases involved in obtaining bus travel-time data, it is considered as the best option as a probe vehicle for other vehicles in the traffic stream. This study includes the modeling bus characteristics such as bus stop dwell time by considering parameters like passenger count and dwell time at individual bus stops.

The traditional travel-time prediction models in the literature are broadly classified into four categories: Historical average models [14–17], Statistical models [18–21], Model-based approaches [22–27] and Machine learning models [28–35]. Among these, Machine learning models are gaining popularity as these advanced models are well suited for supervised and unsupervised learning. This paper involves mode-wise travel-time estimation along an arterial corridor using Machine learning techniques (ANN and SVM).

A suitable study corridor of 59.48 kilometers in length was chosen for this study. In this study corridor, there are 19 bus stops; the positional coordinates (latitude and longitude) of the bus stops were collected using a hand-held GPS during the field survey. The whole corridor was segmented into 18 successive segments based on the 19 successive bus stop locations. The proposed travel-time estimation method is developed in two stages. In the first stage, the segment travel-time data of bus and car along with other parameters such as segment length, average speed at segments, intersection length and signal timings (if the intersection falls within the segment), bus stop dwell time, passenger boarding and alighting (if the bus has stopped at the bus stop within the segment) were given as inputs to the neural network for training. A back-propagation algorithm was used to train the model. In the second stage of development, the data is firstly classified based on the road geometry (number of lanes and gradients). Thus, the complete corridor data was segregated based on lanes -single lane to six lanes; and further, the data were categorized using gradients as a binary classifier in the SVM modeling process. The classified data along with the other parameters (length of the segments, bus travel-time, bus stop details such as halt time and passenger count, intersection length and signal time (red-time) were modeled using the SVM. The model parameters chosen for the ANN and SVM models were optimized using the k-fold cross-validation technique. For mode-wise travel-time modeling, a deeper understanding of the existing prediction models is required. The following sections provide a literature review on existing prediction models used for travel-time prediction.

2. Literature review on travel-time prediction models

The travel-time prediction models can be categorized as statistical and simulation models. Various statistical models are: Regression models [15,36–39], Time series models [40–44], Kalman-filtering models [17,24,32,42,45,46], ANN models [17,30,32,34,47–50], Support Vector Machines[31,34,35,50–54], Hybrid models [29,31,51,55,56]. Among these wide ranges of prediction models, Machine learning models have gained popularity. Machine learning models are capable of performing non-linear modeling; and also these are considered as appropriate techniques for handling complex data. It is evident from the extensive literature that ANN is being widely used for bus arrival time prediction because of its ability to solve complex nonlinear relationships [17,32,34,50]. To account for the nonlinear nature of travel-time data for prediction, Kisgyorgy and Rilett [57] suggested modifications such as clustering techniques, modular neural network, expanded input nodes, and spectral basis neural network to ANN. Chien at al. [42] used integrated adaptive neural network algorithms to identify the prediction error in real-time; the bus travel-time prediction is assessed with a microscopic simulation model (CORSIM). Ishak and Alecsandru [58] used multiple topologies of the dynamic neural network to optimize the short-term travel-time prediction. However, existing studies [28,34,50,53,54] claimed that SVM is advantageous over ANN for short-term prediction. Zhong et al. [12] suggested assigning higher weights to the important variables can reduce the outliers and minimize their impact. Muller et al. [53] suggested that kernel algorithms are more efficient for travel-time prediction. However, the application of an accurate prediction methodology for mode-wise travel-time prediction is considered as a potential research challenge for researchers.

The first work attempted in using public transit as probes for estimating the travel time of other modes was pioneered by Bae et al.[59]. The study used buses as probes for car travel-time estimation. They had used the simple regression and ANN (artificial neural network) methods for analysis and found that the ANN method outperformed the simple regression methods. The study considered only historical data, and real-time prediction was not attempted. Hall and Vyas[60] compared bus probe data with automobile trajectories and the study resulted in analyzing the delays between buses and automobiles are interrelated. The analysis revealed that longer delays of automobiles also resulted in delays on buses traveling on the same route. Bertini and Tantiyanugulchai [61] developed travel-time estimation models between automobiles and buses by eliminating bus stop dwelling time. The study by Bertini et al. [62] compared the time-distance diagrams of different types of bus trajectories with the time-distance diagram of cars. It was found that pseudo bus trajectories are able to explain the car travel-time. But the study was tested for the morning peak period alone.

Most of the research from existing literature have used either linear regression modeling or a combination of two or more methods for mode-wise travel-time prediction. It can be seen that the combination of Kalman filter and ANN were predominantly used in literature. Chakraborthy et al. [36] developed simple regression equations that used buses as probes to determine automobile travel-time. This method was not successful in predicting travel-time in real-time. Padmanabhan et al. [63] estimated the bus travel-time in a linear relationship by incorporating the dwell time into consideration. Kumar et al. [64] on contrary estimated the travel-time for different modes with respect to buses used as the probe vehicle where the bus travel-time was correlated to other modes by removing the dwell times at bus stops with associated acceleration and deceleration. In their study, two methods were proposed: (1) regression analysis and (2) ratio method to analyze mode-wise travel-time estimation. Shalaby and Farhan [65] proposed a bus travel-time prediction model using the AVL (Automatic Vehicle Location) and APC (Automatic Passenger Counter) data to model bus travel-time. The performance of the model was tested using a micro-simulation tool - Vissim. A Kalman filter was fitted to check the accuracy of the model. Jeong and Rilett [48] studied three different models to predict bus arrival time. Historical data-based models, Regression Models, and Artificial Neural Network (ANN) models were used; and it was found that ANN models outperformed the other two. In this research, the AVL data was used and developed a model considering the dwell time and schedule adherence. Essaway et al.[66] used bus travel-time data to estimate general link-travel-times of neighbor (nearby) links. In this study, regression models were developed to relate bus travel-times to general link travel-times. Consequently, the estimated link travel-times were used to calculate the travel-times of neighboring links. A summary of existing studies that have used the probe vehicle technique for travel-time estimation is detailed in Table 1. From Table 1, it is evident that in general regression models, Kalman filters and ANN are commonly used for mode-wise travel-time estimation. Both regression models and Kalman filters can perform linear modeling. But, travel-time is a dynamic parameter and the factors affecting travel-time are also very complex and dynamic in nature. In order to model such complex parameters, developing a non-linear model is required.

In order to estimate the travel-time in the arterial corridor, an appropriate prediction methodology should be considered. The prediction model should be capable of capturing and modeling the relationship between various travel-time affecting parameters for mode-wise conditions. Machine learning models are gaining popularity as these advanced models are well suited for supervised and unsupervised learning. This paper involves mode-wise travel-time estimation along an arterial corridor using Machine learning techniques (ANN and SVM). The main reasons to choose ANN and SVM for mode-wise travel-time prediction are:

(1) They are parametric models and suitable for handling complex data;
(2) These methods are very efficient in modeling and deriving relationships involving bi-modal (bus and car) parameters. Travel-time is one such parameter that is influenced by many other factors such as traffic and roadway characteristics. For example, travel-time prediction at intersections is very challenging as the signal timings are mostly dynamic and actuated; such multi-level parameter modeling can be efficiently performed using machine learning models.
(3) ANN models have the ability to model complex parameters with repetitive iterations using randomized weights and bias values.

SVMs have the capability of generalizing the data and it is also possible to achieve global minima for given training data. Further, ANN and SVM are highly optimized models to approximate the results between the actual and predicted values to some extent.

Table 1
Travel-time estimation using probe vehicle technique.

Author	Technique	Key Findings
Bae [67]	ANN, regression	ANN outperformed regression.
Hall and Vyas [60]	Compared the trajectories of bus and car	The delays of buses and other modes are inter-linked. Delay in automobiles consequently results in the delay of buses in a link.
Cathey and Dailey [22]	Kalman filter	The study resulted in understanding correlations between travel-time obtained from probes and loop detectors.
Tantiyanugulchai and Bertini [61]	Bus and car trajectories	The travel-times of automobiles and buses can be correlated by eliminating bus stop dwelling time.
Chakraborthy and Kikuchi [68]	Regression	Proved buses can be used as probe vehicles to determine automobile travel-time.
Jeong and Rilett [69]	Historical data-based models, regression and artificial neural network	The ANN models outperformed Historical and Regression models. The ANN model was developed considering the dwell time and schedule adherence.
Bin et al. [54]	SVM and ANN	The study travel time using three input variables, results indicated that the SVM model outperformed ANN model
Padmanaban et al. [70]	Kalman filter	Estimated travel time of buses incorporating the bus stop dwell time
Kieu et al. [71]	Regression	The model was suitable for off-peak hours.
Esawey and Sayed [66]	Regression and Vissim	Estimated car travel-time with an error of 17.6%.
Zhan et al. [72]	Link travel-time estimation using MNL model	The link travel-time is estimated by minimizing the error between expected path travel-time and observed path travel-time
Vasantha Kumar and Vanajakshi [64]	Regression and correlation ratio method	Estimated travel time between automobiles and buses by eliminating bus stop dwell time with acceleration and deceleration at bus stops.
Zhou et al.[39]	Regression analysis and frequency distribution	Bus arrival prediction using a smart card system. The model calculated bus arrival time incorporating the passenger alighting and smart card swiping time
Arhin and Stinson [73]	Regression analysis	The number of passengers alighting, passenger boarding, number of access approaches and signalized intersections was identified as significant parameters in bus arrival time estimation
Sharmila et al. [74]	Hybrid model using SVM-PF	Used a bi-modal modeling for buses and cars. Estimated MAPE Value attained was 9.96 (car) and 11.24 (bus)
Kumar et al. [46]	Time-space discretization using Kalman-filter	A speed based traffic stream models was developed using the Godunov scheme. Kalman filter was used for prediction. Kalman filter method outperformed ANN, Regression and Historical average
Zhang et al. [33]	Pattern matching	The Spatial-temporal traffic patterns are matched for multi-step travel-time forecasting.

3. Study corridor

The chosen study corridor for this study consists of busy arterial roads of the Mumbai city covering a stretch of 59.48 kilometers. This study corridor is considerably long connecting the Jogeshwari –Vikhroli link road (10.6 km), the Western Expressway (25.33 km) and the Eastern Expressway (23.55 km) forming a triangular corridor around the IIT Bombay campus. The proposed study corridor contains nineteen bus stops and nine signalized intersections. Figure 1.a. shows the selected study corridor for this study. The data collection process for this study was performed by collecting real-time travel-time data from the field for the two modes- car and bus using onboard hand-held GPS (E-Trex 10) as shown in Figure 1.b. The data were collected for five days (Monday-Friday) for both the modes bus and car for the morning peak period (from 8.00 am to 11.00 am).

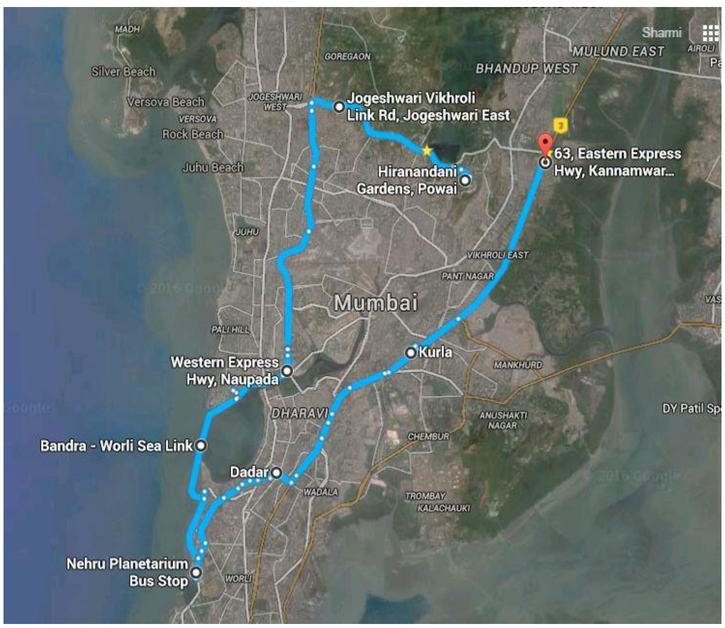

Fig. 1.a. Study corridor.

Fig. 1.b. Data collection using hand-held GPS.

3.1. Study parameters

In order to achieve car travel-time using the bus as probes, enormous factors were considered for modeling the relationship between them. Out of which certain important parameters were grouped into two main categories such as traffic parameters (distance, speed, travel-time, intersection length) and road-way characteristics (road width, gradients bus stop halt time, passenger count, etc.). Table 2 gives the details of the variables used in this study, and these parameters were selected based on literature review and field conditions adaptable to the study corridor.

Table 2.
List of variables affecting travel-time and their descriptions

Variable	Description
Bus stop based segments	The study corridor was segmented using the identified bus stop waypoints.
Distance	The haversine formula was used to calculate the distance of the links/segments with the help of the latitude and longitude positions obtained from the GPS data.
Car travel-time	Segment travel time data for the car was obtained using the GPS enabled in the car.
Bus travel-time	The segment travel-time for the buses was calculated using the handheld GPS device during data collection.
Intersection length	The intersections on the route were marked on the field, and their length was determined from the field and compared with the Google distance calculator to check for the accuracy of the measured intersection length and stop time for the red cycle at various intersections was determined from the field during the night survey conducted.
Gradients	Using the elevation information obtained from the GPS data the gradients were estimated.
Lanes	The number of lanes was identified throughout the corridor during the field study, and it was observed that the corridor contains a single lane to six lanes at different stretches of the road. The length of the road stretch for every lane was calculated.
Halt time	The bus dwell time was manually counted at each time the bus stopped at bus stops using hand-held GPS and stopwatch
Passenger Count	The number of passengers boarding and alighting the bus at every bus stop was also noted during the survey.

The above-listed parameters in the table are chosen as the input parameters for both ANN and SVM modeling. These parameters are considered as the most important attributes which affect corridor level travel-time estimation.

4. Methodology

This study is oriented in determining the travel-time of different modes on the arterial corridor using buses as the probe vehicles. The idea of mode-wise travel time estimation is very less used due to the various complications involved in modeling when considered for two different modes. Therefore, this study aims at modeling mode-wise travel time by including all the relevant parameters that are influential with respect to the two modes (bus and car). Fig.2 illustrates the overall methodological framework of the proposed model. The basic interpretation of the proposed study is given as follows:

$$CTT = f(BTT, IL, Gr, DWT, PC, SPV, SG) \tag{1}$$

Where,

Desired output: CTT= Car travel time; Inputs for the model: BTT= Bus travel time; IL = Intersection length; Gr = Gradients; DWT: Dwell time; PC = Passenger count; SPV = Speed variation

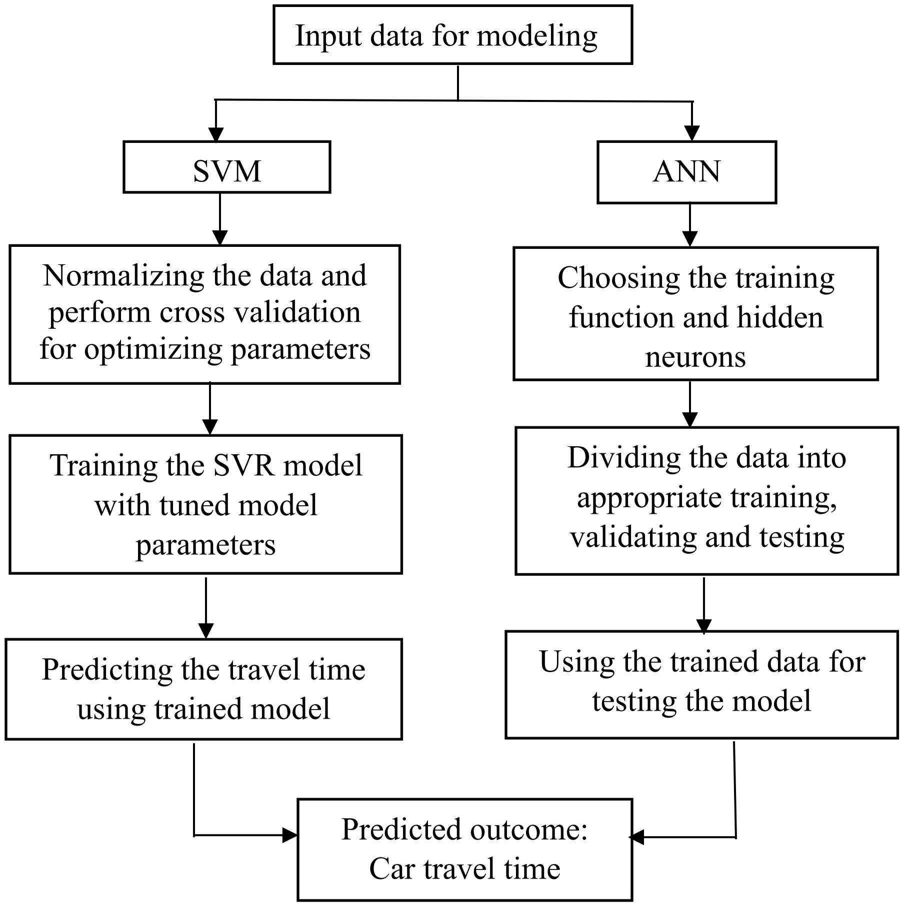

Fig.2. Overall methodological framework

4.1. ANN model development

Although the basic training procedures of ANNs are almost similar, the accuracy of the result is greatly dependent on the type of input/output combinations.

An artificial neural network mathematical model is written as:

$$y = f(U) = W_0 * \tanh(W_i * U + B_i) + b_0 \tag{2}$$

Where y is the output of the neural network model,

U is a column vector of size p that contains the p inputs of the process;

Wo is a row vector of size n that contains the weights of the neural network model from the hidden layer to the output;

Wi is a matrix that contains the weights of the neural network model from the inputs to the hidden layer. This matrix has n rows and p columns;

Bi is a column vector of size n that contains the biases from the input to the hidden layer of the neural net model;

bo is the bias (scalar) from the hidden layer to the output of the neural net model; and

tanh is the activation function (in this study hyperbolic tangent function is used).

The activation functions are needed to introduce the nonlinearity into the network. They determine the non-linear relationship between the input and the output layers. There are many activation functions used in practice (for example, piecewise linear, Gaussian and sigmoidal functions). The sigmoidal functions such as logistic and hyperbolic tangent functions (tan h) are the most common choices. Sigmoidal functions such as tanh or arc tan produce both positive and negative values tend to yield faster training than the other functions such as logistic functions that produce only positive values [75]. Hence, in this study, the tanh function is used to scale inputs and targets to (-1, 1).

The activation function for tanh function is given by the following formula:

$$f(\mathrm{x}) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

The curve for tanh is very similar to sigmoid with the only real difference lies in output range for both activation functions. The tanh function can map input values to a range between -1 and 1. It is centered at 0. Like the logistic sigmoid, the tanh function is also sigmoidal (“s”-shaped), but instead outputs values that range (-1, 1) Thus strongly negative inputs to the tanh will map to negative outputs. Additionally, only zero-valued inputs are mapped to near-zero outputs.

4.2. Training procedure: back propagation algorithm

Backpropagation is a powerful algorithm that is used to train the multilayer perceptron and obtain the weight of each link. These synaptic weights enable the network to move closer to the desired response by frequent iterations. A most common measure of the error in the backpropagation is the mean square error.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - y_0)^2 \tag{4}$$

Where y_i is the predicted value and y_0 is the observed value

The backpropagation algorithm is the most popular algorithm for transportation use[14] [16]. Hence, in this study, authors have resorted to using the backpropagation algorithm.

4.3. Network architecture

Existing literature consists of different types of ANN architectures for forecasting purposes. However, the multi-layer perceptron has gained popularity among all other structures [75]. In this study, an optimal ANN model was developed using different combinations of network architecture. The ANN (i,j,k) indicates a network architecture with i, j and k neurons in the input, hidden and output layers respectively. The hidden and output layers are responsible for the actual processing in the network. As part of the training process, the weights and bias parameters are generated in the hidden layer. The nodes of the hidden layer capture the pattern of the data and perform a nonlinear mapping between the input and output variables. The MATLAB tool was used for determining the training sets, training ANNs and initializing the values of weights and biases using trial and error approach and optimizing for the best fit model. Table 3 and Table 4 show the weights and biases values of the trained networks.

Table 3
Weights of the trained network.

$w_{i,j}^{k}$	K	J								
i		1	2	3	4	5	6	7	8	9
1	1	1.448	1.683	-1.75	1.56	1.031	-1.76	-1.41	0.4993	2.005
	2	2.139	-1.42	2.209	1.725	-0.11	1.317	-2.27	-0.833	1.822
	3	1.344	1.905	0.731	-1.72	2.720	1.904	1.13	2.745	1.065

$w_{i,j}^{k}$ is the weight between j^{th} neuron of i^{th} layer and of the k^{th} neuron of the previous layer

Table 4
The bias of the trained network.

B_i^j	J								
i	1	2	3	4	5	6	7	8	9
1	-2.91	-2.18	1.456	-0.72	0	-0.72	-1.45	2.184	2.912
2	0.03	-1.5	-0.29	-1.18	0.07	-0.28	-0.12	-0.14	0.006

B_i^j is the bias of j^{th} neuron of the i^{th} layer

The weights and bias values were estimated as deliverables of the ANN modeling using the following equations:

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i \tag{5}$$

w are the weights; α_i: Lagrange multipliers; x_i are the inputs and the bias values are calculated as:

$$b = \frac{1}{S}\sum_{i=1}^{S} (\mathrm{y}_i - w.x) \tag{6}$$

Where S is the support vectors.

4.4. Data modelling using ANN

In this study, authors are attempting to predict car travel-time using the buses as probe vehicles. It is appropriate to segregate the data as segments based on bus stop locations to understand the variation between the travel-time of buses and other modes (car). Hence for this study, bus stop locations were identified using GPS in the field. The GPS data for cars and buses were segmented based on these bus stop locations. There are 19 bus stops within the study corridor; which implies that there are 18 segments in total for ANN modeling. The input parameters considered for the ANN modeling are (1) length of the segments, bus travel-time, and bus stop details such as halt time and passenger count, intersection length and cycle time (red-time). The car travel-time is the desired output of the ANN model. These parameters were modeled in ANN using MATLAB software. The complete five days of segmented data were used as a training and testing data for the ANN model containing the input parameters as described above. The model used 80 % of the data for training purposes and the remaining data for testing and validation [17,54]. The number of neurons used for the model was 20 (optimized after several iterations). A backpropagation algorithm was used for training the model. The number of epochs used for this model was 500 (maximum limit obtained in the model). Car travel-time was obtained as the predicted outcome of the model. Fig.3. illustrates the details of the ANN algorithm and its specifications.

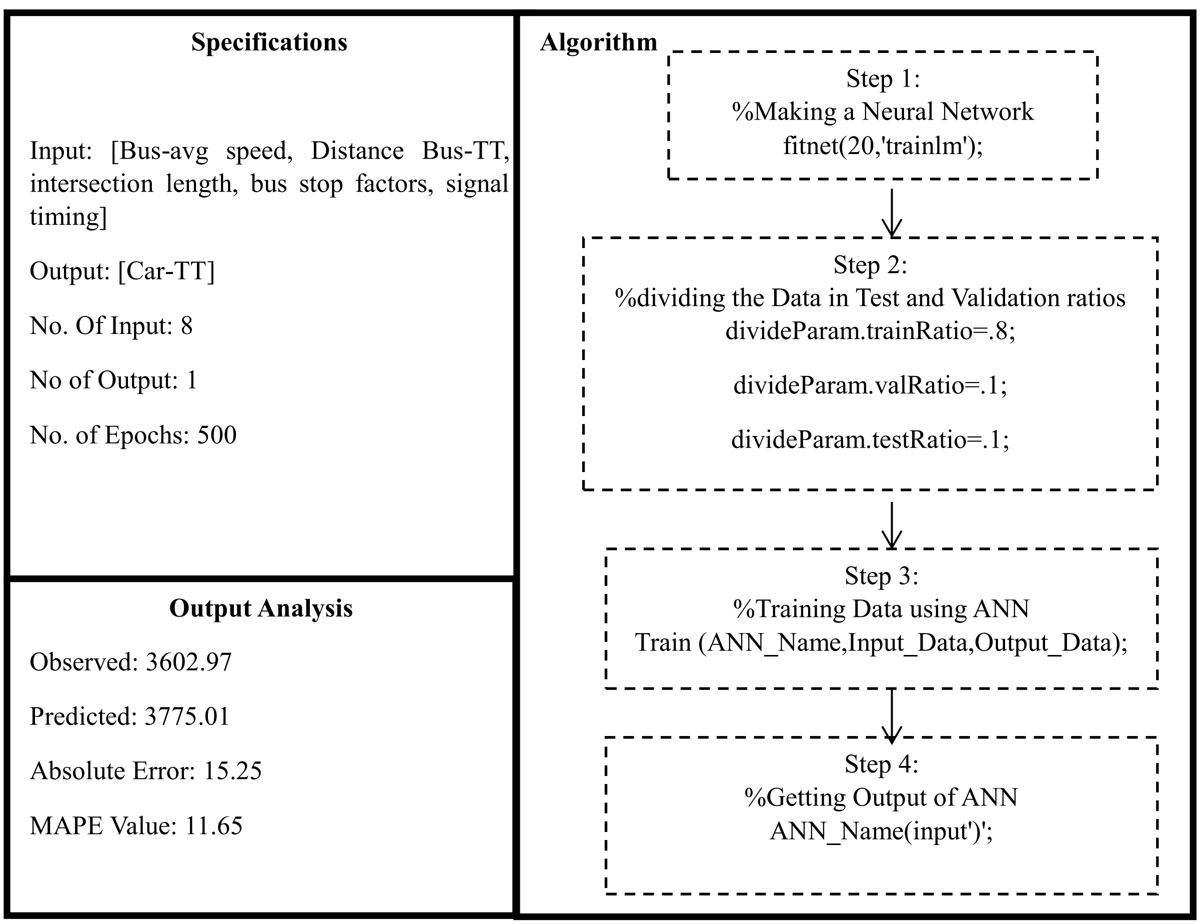

Fig. 3. ANN algorithm and specifications.

4.5. Support vector machine: mathematical formulation

SVMs are learning systems that generalize on a hypothetical space of linear functions on a higher dimensional feature space using a trained learning algorithm [76]. The SVMs are used in fitting a hyper-plane to the training data through which the data is classified into two classes. The hyperplane acts as a decision surface as a margin of separation between the positive and negative classes. With these classes, the hyper-plane is obtained by maximizing the margin between the hyper-plane and the closest training samples, called the "Support vectors" [77]. The support vector machines work on the principle that the data points which are non-linearly separable are transferred to a higher dimension (D) with the help of kernel functions to make them linearly separable in a higher dimension scale. In the higher dimension, a hyperplane is constructed between data points to maximize the margin of separation. For additional details on the general concept of SVM, see Vapnik [78], Burges [79], and Scholkopf and Smola [80].

In a linear SVM model, the general SVM equation is given as:

$$f(\mathrm{x}) = \mathrm{w}.\mathrm{x} + \mathrm{b} \tag{7}$$

where w is weight vector and b is bias. f(x) is the function associated with the hyperplane.

Let us assume the training data D has a set of n points as given by [77].

$$D = \{(\mathrm{x}_i, y_i) \big| x_i \in R^d, y_i \in \{-1, +1\}_{i=1}^{n} \tag{8}$$

y_i belongs to -1 or 1, to which the point x_i belongs to in a d- dimensional feature space, R^d .

In cases, where the data points are not linearly separable, Cortes and Vapnik [81], using a modified SVM algorithm by adding a soft margin [77]. A sample form of the SVM function in a higher dimensional space, as illustrated in [76]:

$$\hat{y}(\bar{x}, \bar{w}) = \sum_{i=1}^{n} w_i \phi_i(x) + w_0 = \bar{w}\phi(x) + w_0 \tag{9}$$

Where $\phi(\mathrm{x})$ represents the high dimensional feature spaces. By adopting the Lagrangian multiplier method, the parameters such as w_0 and $\bar{w}$ can are estimated. The details of optimization can be obtained

$$\min_{w,b,\varepsilon}.\{\frac{\|w\|^2}{2} + C\sum_{i=0}^{n} \xi_i\} \tag{10}$$

subject to $y_i(w.x_i + b) \geq 1 - \xi_i$ for i= 1……………………, N

The factor C and the slack variable ξ_i in the previous equation

The regularization parameter C takes into account the mislead points and maintains the shape of the function and the slack variable ξ_i measures the degree of misclassification of the data points in x_i.

Finally, the maximum margin in the hyperplane after Lagrange minimization is given as below:

$$f(x) = \sum_{i\varepsilon n} \alpha_i y_i k(x_i, x_j) + b \tag{11}$$

Where, $k(x_i, x_j)$ is the kernel function and α_i is the Lagrange multipliers, n= set of support vectors.

In the cases where the data is not linearly separable, the kernel trick is applied to transform the feature space to a higher dimension space where the data is linearly separable.

One common example of kernel function is the Gaussian radial basis function $yK(\mathrm{x}_i, x_j)$ given by Rakshita et al. [82]:

$$yK(\mathrm{x}_i, x_j) = \exp(-\frac{\|x_i - x_j\|}{2\delta^2}) \tag{12}$$

Where, δ^2 is the bandwidth of the kernel. In the present study, since the classes were overlapping, we used a soft-margin SVM with RBF kernel.

4.6. Hyper-plane creation criteria

The GPS data was classified based on the road geometrics such as a number of lanes and gradients. The complete corridor data was segmented based on the change of lanes observed throughout the study corridor. As per the field observations, it was identified that the whole corridor consists of single, two lanes, three lanes, four lanes, five lanes, and six lanes. Hence the study corridor was segmented based on the changes in the road width during the vehicle movement and the GPS data of bus and car was segregated based on lanes. Also, the gradients on the road were also determined through the elevation data obtained from GPS. Later the gradient of the road was used as a binary classifier to train the model using SVM. The negative gradients were labeled as 0 and the positive gradients were labeled as 1. This binary classification helped the SVM model to classify the data based on the gradients, and a hyper-plane was created to form the boundary between the negative and positive gradients.

4.7. Model development using SVM

In the second stage of modeling the data using SVM, a k-fold cross-validation technique was used in order to optimize the essential parameters and minimize the bias between the training and testing data. For this study, a radial basis function was used as a kernel function. The gamma g, cost function c and epsilon values taken for the model are g=2, c=256, e=0.1. The gamma and cost function parameters used in the model were optimized using the k-fold cross-validation technique. The predicted output of the model is the car travel-time. The selection of parameters for the model is an important step in deciding the model's accuracy. The performance and consistency of the model highly depend on the model parameters used in the model. For this study, the data was validated in two stages; initially, authors have performed analysis and validated data using the normal train/test split method (dividing the entire dataset into two sets; one training set and another testing set). The advantages of using this method are: (1) it trains faster than other existing methods like cross-validation etc.; (2) also, it is simpler to examine the detailed results of the testing process. However, this method does not account for the biases involved in the splitting of training and testing data. Therefore, to bring homogeneity in results for the model developed, the training and testing data was validated using the k-fold cross-validation procedure. The five days data was taken for training the model, and the k-fold cross-validation method was used in selecting the optimum parameters (Cost function and gamma). Kernel functions used in SVM modeling are responsible for the mapping of input data into feature space. Popular choices of kernel function include radial basis function (RBF), linear kernel, and a polynomial kernel function. However, in this study RBF kernel is used; because RBF is highly effective in mapping nonlinear relationships. There are two parameters for an RBF kernel: Cost function (C) and Gamma (γ). The C and Gamma values are usually chosen as user perceived values. There is no standard method available to determine the optimum values for the C and gamma parameters. Thus, it is essential to identify a method to optimize the parameters such that the model classifier can accurately predict unknown data (i.e., testing data). For solving this problem, common strategies (known as k-fold cross-validation) were used in this study. In the k-fold cross-validation, as the first step, the training data was divided into k subsets of equal size. Sequentially, one subset is tested using the classifier trained on the remaining k-1 subsets. The complete set of five days data was divided into five equal partitions (i.e., k equal to 5). Out of five days of data, four days of data were used as training data (Day 2, 3, 4, and 5) and the remaining one-day data was used as the testing data (Day 1). Then the least error was calculated for each segment of the day -1. The optimum C and Gamma values were obtained from the k-fold cross-validation technique. A detailed algorithm with specifications of the SVM model is given in Figure 4.

5. Results and discussions

The results obtained from the developed ANN and SVM algorithms were compared with the observed travel-time data obtained from the field. The prediction was carried out from the five days of data collected for both bus and car modes. After the data is trained using the ANN and SVM models, then the next set of data was used for validation; and the output derived is the predicted data obtained as the result of the trained model. In this paper, authors have used the bus and car data as inputs for training and obtained the predicted results of car data with respect to buses used as probe vehicles. The prediction accuracy was measured in terms of the MAPE (Mean Absolute Percentage Error), MAD (Mean Absolute Deviation) and RMSE (Root Mean

Square Error) values. Figures 5 and 6 show the prediction results for ANN and SVM in comparison with the original field observed values.

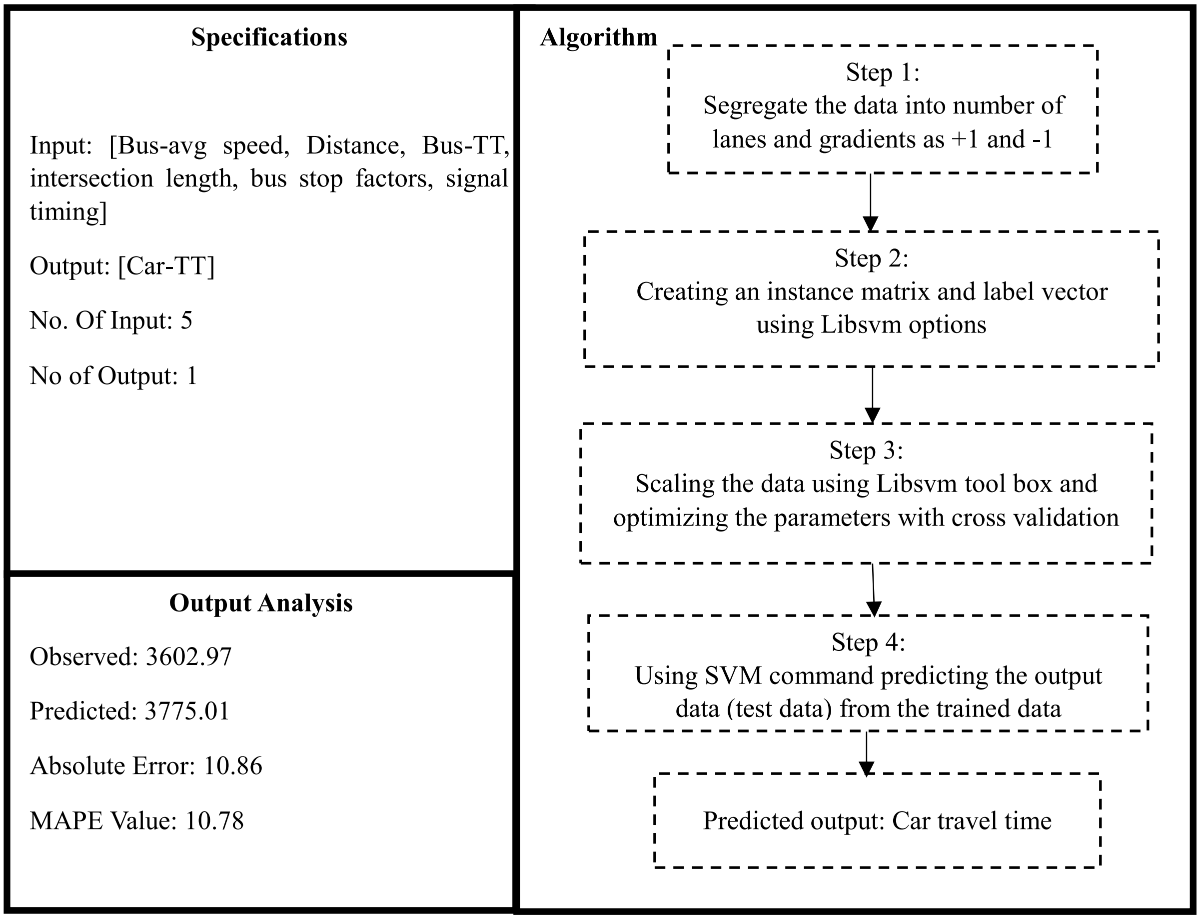

Fig. 4. Support Vector Machines algorithm and specifications.

Mean absolute percentage error (MAPE)

The two models were evaluated based on the MAPE, MAE and RMSE values

$$MAPE\ value = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_t - y^t}{y_t}\right)\times 100 \tag{13}$$

y_t – The observed value, y^t - predicted value, n number of observations.

Mean average deviation (MAD)

$$MAE = \frac{\sum_{i=1}^{N}[\![x_{pred} - x_{obs}]\!]}{N} \tag{14}$$

x_{pred} is the predicted value; x_{obs} is the observed value; N is the total number of segments

Root mean square error (RMSE)

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_i - y_t)^2}{n}} \tag{15}$$

y_i is the observed value; y_t is the predicted value and n is the total number of segments

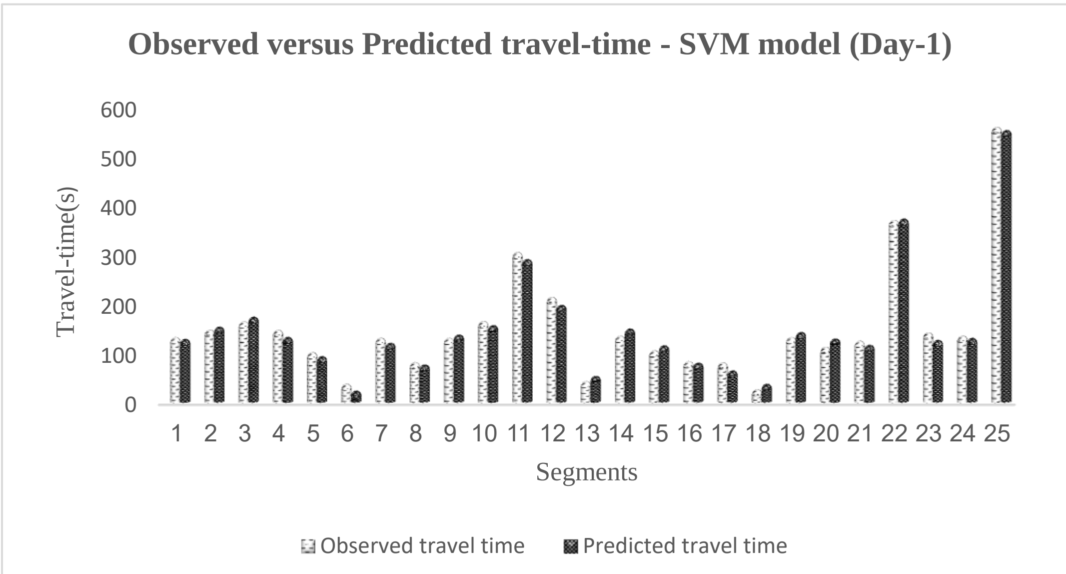

Fig. 5. Observed versus predicted travel-time: SVM model.

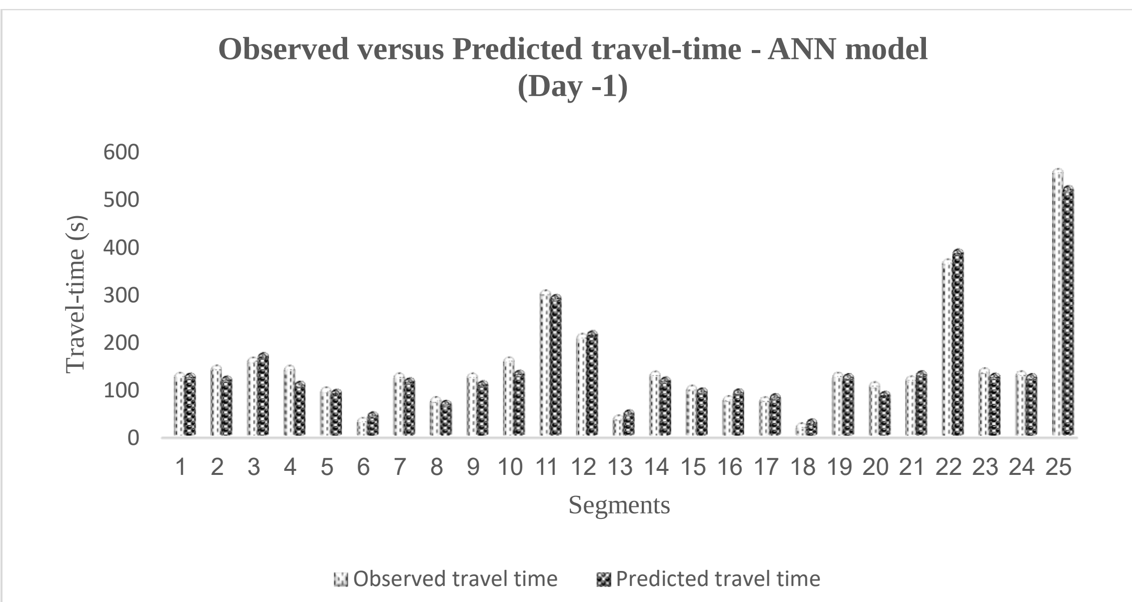

Fig. 6. Observed versus predicted travel-time: ANN model.

A weighted percentage error graph for the models developed (ANN and SVM) are plotted in Figure 7. Also, a comparison plot of the speed profiles for the different days is given in Figure 8.

Table 5 gives the statistical accuracy measures for the predicted data for the ANN and SVM models. The exact percentage accuracy of the model was tested using very common error measures namely, the mean absolute deviation, root mean square error and the mean absolute percentage error.

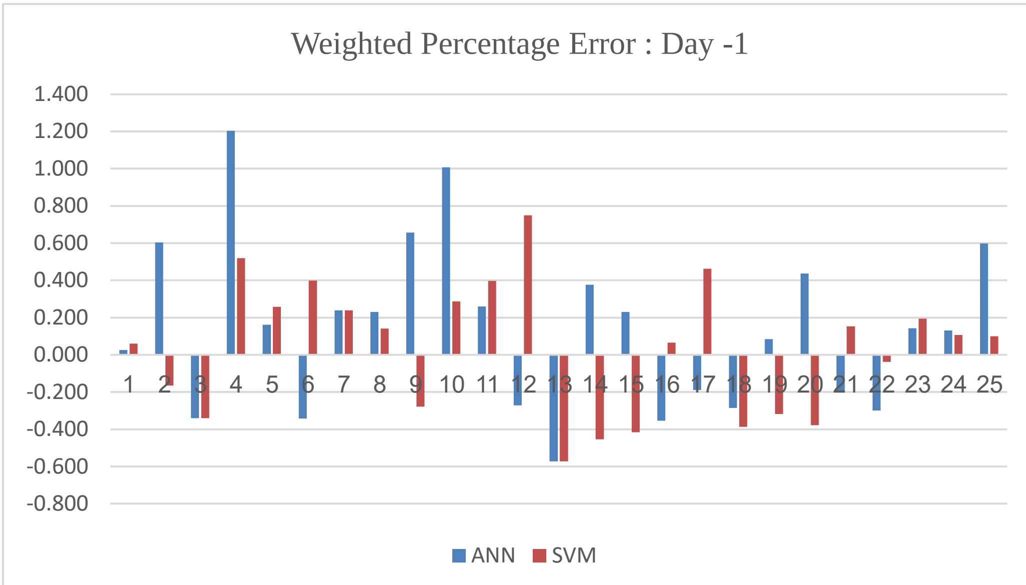

Fig. 7. Weighted percentage error graph

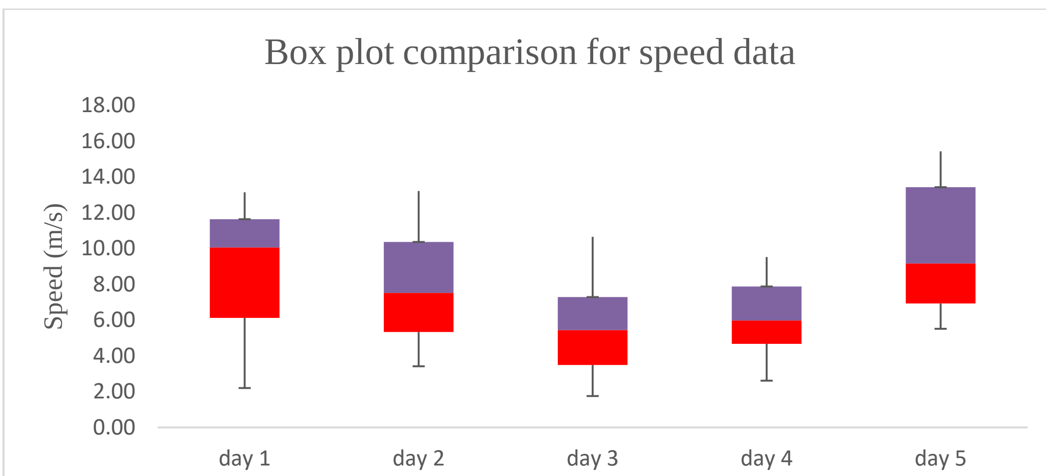

Fig. 8. Comparison graph for speed profile

Table 5
Measures of percentage accuracy for the model developed.

A measure of accuracy (Day -1)	ANN model	SVM model
MAPE (%)	11.65	10.78
MAD (s)	13.06	9.85
RMSE (s)	15.8	10.8
Standard error (s)	15.25	10.86

It is inferred from Figures 5 that the actual values and the predicted values are almost close except for certain locations (segments) where the deviation in the values may be due to varying traffic congestion at a different location on different days. The MAPE values for ANN and SVM models are identified as 11.65 and 10.86 respectively. According to Lewis' scale of interpretation of estimation accuracy [83], any forecast with a MAPE value of less than 10% can be considered highly accurate, 11–20% as good, 21–50% as reasonable, and 51% or more as inaccurate. Thus, the results obtained are reasonable for all with MAPE of less than 15%.

6. Statistical validation of the proposed models using Kolmogorov-Smirnov test (K-S test)

The Kolmogorov-Smirnov test is used as a statistical evaluation tool to compare the observed sample distribution and theoretical sample distribution. The cumulative distribution function for a variable with a specified distribution can be compared using this test. It was observed from chi-square and joint t-tests that the travel-times follow the normal distribution. Here in this study, the Kolmogorov-Smirnov test is used as a tool to evaluate the differences in the cumulative distribution for the observed and the predicted values obtained from the ANN and SVM models. The null hypothesis assumes no difference between the observed and theoretical distribution.

Acceptance Criteria: If the calculated value is less than the critical value, accept the null hypothesis.

Rejection Criteria: If the calculated value is greater than the table (critical) value, reject the null hypothesis.

The test results of the Kolmogorov-Smirnov test conducted for the observed and predicted values are depicted in the form of a cumulative frequency graph in Figures 9 and 10. Table 6 gives the test results for the Kolmogorov-Smirnov test conducted.

Table 6
Kolmogorov Smirnov (K-S) test.

Kolmogorov Smirnov test	Calculated value	Tabulated value	n value	Alpha value	Null hypothesis
ANN model	0.028142	0.309	18	0.05	Accepted
SVM model	0.047011	0.375	12	0.05	Accepted

The max D value obtained for cumulative differences between the observed and predicted values for the ANN model is 0.028142 and the critical value obtained from the table is 0.309 (with a degree of freedom as 18 and alpha value as 0.05). Similarly, for the SVM model, the cumulative differences obtained are 0.047011 and the critical value obtained from the table is 0.375 (with a degree of freedom as 12 and alpha value as 0.05). It is observed that for both the models, the calculated values are much lesser than the critical (table) values indicating that there are no major variations between the observed and predicted travel-times at a 95% confidence level. Hence the null hypothesis is accepted.

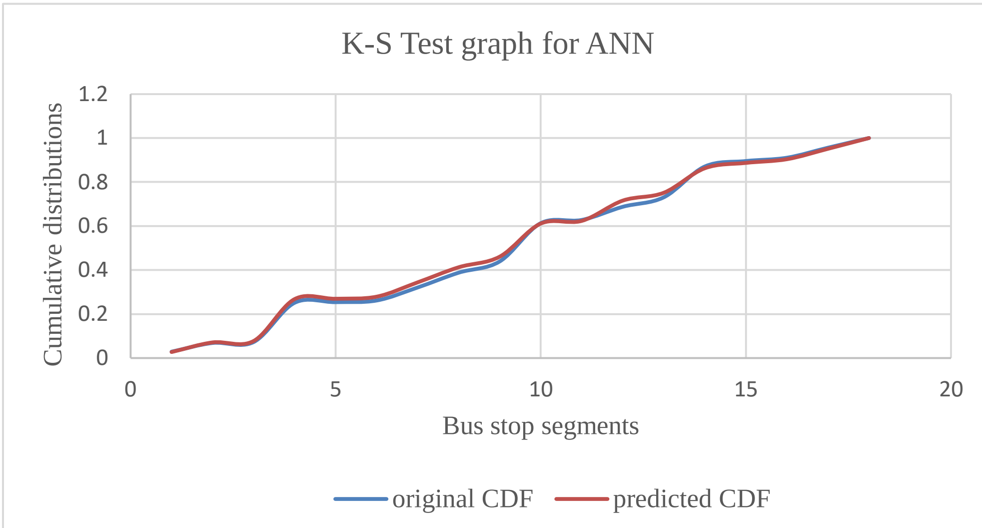

Fig. 9. Cumulative differences graph for the K-S test for ANN model

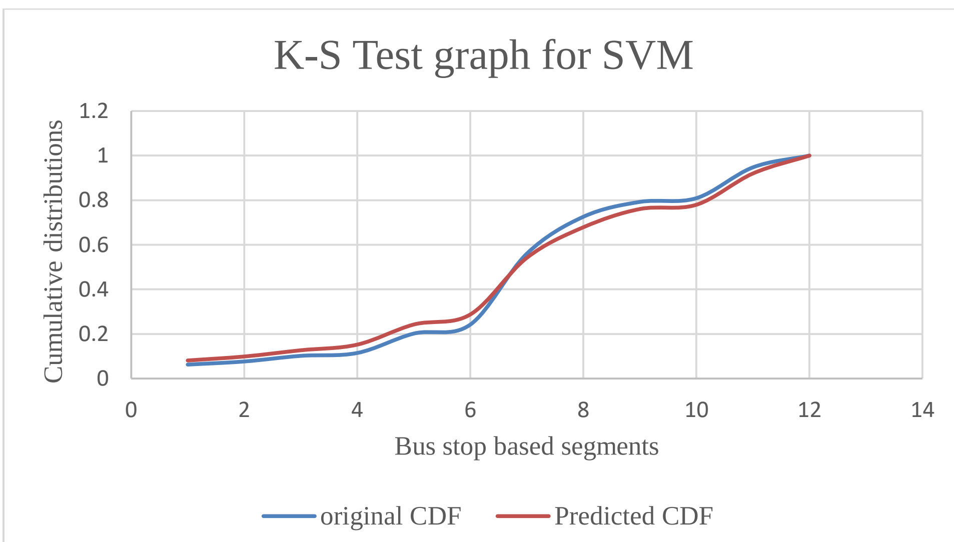

Fig. 10. Cumulative differences graph for the K-S test for SVM model.

7. Summary and conclusions

The present study carried out a detailed analysis of travel-time and it's affecting parameters for mode-wise travel- time estimation. This study was conducted with a motive to predict mode-wise travel-time as it is considered advantageous for collecting individual data for all the modes. Therefore, this method solves the purpose of extensive data collection, which is tedious for a heterogeneous traffic condition. In this paper, authors have attempted to find the car travel-time considering buses as probe vehicles using ANN and SVM models. It is evident from the results that corridor level travel-time can be estimated more efficiently by segmenting the corridor into bus stop sections and modeling using the Machine learning models. From the results, it can be seen that the ANN and SVM models give a reasonable MAPE value of 11.65 and 10.78 respectively. It is observed that the obtained values are much lower compared to the models developed for travel-time estimation for Indian conditions. The significance of this study is modeling the travel-time affecting parameters and deriving a non-linear relationship model between the two modes. This study includes certain mode characteristics such as bus stop dwell time, passenger count, etc. The possibility of applying machine learning models and deriving an optimized parameter modeling is one of the research highlights of this study. Also in this study, the support vector machines are used as both classification and prediction tools. Thus, these models developed are highly optimized and efficiently used for mode-wise prediction under Indian traffic conditions. This study also includes the statistical evidence for validating the proposed model using the Kolmogorov-Smirnov test. It is observed that there are no major variations between the observed and predicted travel-times at a 95% confidence level.

References

[1] Konbattulwar V, Velaga NR, Jain S, Sharmila RB. Development of in-vehicle noise prediction models for Mumbai Metropolitan Region, India. J Traffic Transp Eng (English Ed 2016;3:380–7. doi:10.1016/j.jtte.2016.04.002.

[2] Hess DB. Access to Employment for Adults in Poverty in the Buffalo-Niagara Region. Urban Stud 2005;42:1177–200. doi:10.1080/00420980500121384.

[3] Kawabata M. Job Access and Employment among Low-Skilled Autoless Workers in US Metropolitan Areas. Environ Plan A Econ Sp 2003;35:1651–68. doi:10.1068/a35209.

[4] Kawabata M, Shen Q. Commuting Inequality between Cars and Public Transit: The Case of the San Francisco Bay Area, 1990-2000. Urban Stud 2007;44:1759–80. doi:10.1080/00420980701426616.

[5] Levinson DM. Accessibility and the journey to work. J Transp Geogr 1998;6:11–21. doi:10.1016/S0966-6923(97)00036-7.

[6] Silva C, Pinho P. The Structural Accessibility Layer (SAL): Revealing how Urban Structure Constrains Travel Choice. Environ Plan A Econ Sp 2010;42:2735–52. doi:10.1068/a42477.

[7] Kwok RCW, Yeh AGO. The Use of Modal Accessibility Gap as an Indicator for Sustainable Transport Development. Environ Plan A Econ Sp 2004;36:921–36. doi:10.1068/a3673.

[8] Salonen M, Toivonen T. Modelling travel time in urban networks: comparable measures for private car and public transport. J Transp Geogr 2013;31:143–53. doi:10.1016/j.jtrangeo.2013.06.011.

[9] Lei TL, Church RL. Mapping transit-based access: integrating GIS, routes and schedules. Int J Geogr Inf Sci 2010;24:283–304. doi:10.1080/13658810902835404.

[10] Liu S, Zhu X. Accessibility Analyst: An Integrated GIS Tool for Accessibility Analysis in Urban Transportation Planning. Environ Plan B Plan Des 2004;31:105–24. doi:10.1068/b305.

[11] Moniruzzaman M, Páez A. Accessibility to transit, by transit, and mode share: application of a logistic model with spatial filters. J Transp Geogr 2012;24:198–205. doi:10.1016/j.jtrangeo.2012.02.006.

[12] O'Sullivan D, Morrison A, Shearer J. Using desktop GIS for the investigation of accessibility by public transport: an isochrone approach. Int J Geogr Inf Sci 2000;14:85–104. doi:10.1080/136588100240976.

[13] Mavoa S, Witten K, McCreanor T, O'Sullivan D. GIS based destination accessibility via public transit and walking in Auckland, New Zealand. J Transp Geogr 2012;20:15–22. doi:10.1016/j.jtrangeo.2011.10.001.

[14] Jeong R, Rilett R. Bus arrival time prediction using artificial neural network model. Proceedings. 7th Int. IEEE Conf. Intell. Transp. Syst. (IEEE Cat. No. 04TH8749), IEEE; 2004, p. 988–93.

[15] Ramakrishna Y, Ramakrishna P, Lakshmanan V, Sivanandan R. Bus travel time prediction using GPS data. Proc Map India 2006.

[16] Chien SI-J, Kuchipudi CM. Dynamic Travel Time Prediction with Real-Time and Historic Data. J Transp Eng 2003;129:608–16. doi:10.1061/(ASCE)0733-947X(2003)129:6(608).

[17] Fan W, Gurmu Z. Dynamic Travel Time Prediction Models for Buses Using Only GPS Data. Int J Transp Sci Technol 2015;4:353–66. doi:10.1016/S2046-0430(16)30168-X.

[18] Rajbhandari R. Bus arrival time prediction using stochastic time series and Markov chains 2004.

[19] Suwardo W, Napiah M, Kamaruddin I. ARIMA models for bus travel time prediction. J Inst Eng Malaysia 2010:49–58.

[20] Rahmani M, Jenelius E, Koutsopoulos HN. Route travel time estimation using low-frequency floating car data. 16th Int. IEEE Conf. Intell. Transp. Syst. (ITSC 2013), IEEE; 2013, p. 2292–7. doi:10.1109/ITSC.2013.6728569.

[21] Jenelius E, Koutsopoulos HN. Travel time estimation for urban road networks using low frequency probe vehicle data. Transp Res Part B Methodol 2013;53:64–81. doi:10.1016/j.trb.2013.03.008.

[22] Cathey FW, Dailey DJ. A prescription for transit arrival/departure prediction using automatic vehicle location data. Transp Res Part C Emerg Technol 2003;11:241–64. doi:10.1016/S0968-090X(03)00023-8.

[23] Wall Z, Dailey DJ. An algorithm for predicting the arrival time of mass transit vehicles using automatic vehicle location data. 78th Annu. Meet. Transp. Res. board, Citeseer; 1999, p. 1–11.

[24] Shalaby A, Farhan A. Prediction Model of Bus Arrival and Departure Times Using AVL and APC Data. J Public Transp 2004;7:41–61. doi:10.5038/2375-0901.7.1.3.

[25] Chen H, Rakha HA. Real-time travel time prediction using particle filtering with a non-explicit state-transition model. Transp Res Part C Emerg Technol 2014;43:112–26. doi:10.1016/j.trc.2014.02.008.

[26] Dhivyabharathi B, Anil Kumar B, Vanajakshi L, Panda M. Particle Filter for Reliable Bus Travel Time Prediction Under Indian Traffic Conditions. Transp Dev Econ 2017;3:13. doi:10.1007/s40890-017-0043-z.

[27] Bi J, Chang C, Fan Y. Particle Filter for Estimating Freeway Traffic State in Beijing. Math Probl Eng 2013;2013:1–6. doi:10.1155/2013/382042.

[28] Vanajakshi L, Rilett LR. Support Vector Machine Technique for the Short Term Prediction of Travel Time. 2007 IEEE Intell. Veh. Symp., IEEE; 2007, p. 600–5. doi:10.1109/IVS.2007.4290181.

[29] Hofleitner A, Herring R, Bayen A. Arterial travel time forecast with streaming data: A hybrid approach of flow modeling and machine learning. Transp Res Part B Methodol 2012;46:1097–122. doi:10.1016/j.trb.2012.03.006.

[30] Zheng F, Van Zuylen H. Urban link travel time estimation based on sparse probe vehicle data. Transp Res Part C Emerg Technol 2013;31:145–57. doi:10.1016/j.trc.2012.04.007.

[31] Li L, He S, Zhang J, Ran B. Short-term highway traffic flow prediction based on a hybrid strategy considering temporal-spatial information. J Adv Transp 2016;50:2029–40. doi:10.1002/atr.1443.

[32] Wang L, Zuo Z, Fu J. Bus arrival time prediction using RBF neural networks adjusted by online data. Procedia-Social Behav Sci 2014;138:67–75.

[33] Zhang Z, Wang Y, Chen P, He Z, Yu G. Probe data-driven travel time forecasting for urban expressways by matching similar spatiotemporal traffic patterns. Transp Res Part C Emerg Technol 2017;85:476–93. doi:10.1016/j.trc.2017.10.010.

[34] Yu B, Lam WHK, Tam ML. Bus arrival time prediction at bus stop with multiple routes. Transp Res Part C Emerg Technol 2011;19:1157–70. doi:10.1016/j.trc.2011.01.003.

[35] Akter S, Huda T, Nahar L, Akter S. Travel time prediction using support vector machine (SVM) and weighted moving average (WMA). Int J Eng Res Technol 2015;4:496–503.

[36] Chakroborty P, Kikuchi S. Using Bus Travel Time Data to Estimate Travel Times on Urban Corridors. Transp Res Rec J Transp Res Board 2004;1870:18–25. doi:10.3141/1870-03.

[37] Patnaik J, Chien S, Bladikas A. Estimation of bus arrival times using APC data. J Public Transp 2004;7:1.

[38] Chang H, Park D, Lee S, Lee H, Baek S. Dynamic multi-interval bus travel time prediction using bus transit data. Transportmetrica 2010;6:19–38. doi:10.1080/18128600902929591.

[39] Zhou Y, Yao L, Chen Y, Gong Y, Lai J. Bus arrival time calculation model based on smart card data. Transp Res Part C Emerg Technol 2017;74:81–96. doi:10.1016/j.trc.2016.11.014.

[40] Al-Deek HM, D'Angelo MP, Wang MC. Travel time prediction with non-linear time series. Fifth Int. Conf. Appl. Adv. Technol. Transp. Eng., 1998.

[41] Stathopoulos A, Karlaftis MG. A multivariate state space approach for urban traffic flow modeling and prediction. Transp Res Part C Emerg Technol 2003;11:121–35. doi:10.1016/S0968-090X(03)00004-4.

[42] Chien SI-J, Ding Y, Wei C. Dynamic Bus Arrival Time Prediction with Artificial Neural Networks. J Transp Eng 2002;128:429–38. doi:10.1061/(ASCE)0733-947X(2002)128:5(429).

[43] Xia J, Chen M, Huang W. A multistep corridor travel-time prediction method using presence-type vehicle detector data. J Intell Transp Syst Technol Planning, Oper 2011;15:104–13. doi:10.1080/15472450.2011.570114.

[44] Chen P, Ding C, Lu G, Wang Y. Short-term traffic states forecasting considering spatial-temporal impact on an urban expressway. Transp Res Rec 2016;2594. doi:10.3141/2594-10.

[45] Park D, Rilett LR. Forecasting Freeway Link Travel Times with a Multilayer Feedforward Neural Network. Comput Civ Infrastruct Eng 1999;14:357–67. doi:10.1111/0885-9507.00154.

[46] Kumar BA, Vanajakshi L, Subramanian SC. Bus travel time prediction using a time-space discretization approach. Transp Res Part C Emerg Technol 2017;79:308–32. doi:10.1016/j.trc.2017.04.002.

[47] Cherrett T, Mcleod F, Bell H, McDonald M. Journey time estimation using single inductive loop detectors on non-signalised links. J Oper Res Soc 2002;53:610–9. doi:10.1057/palgrave.jors.2601348.

[48] Jeong R, Rilett R. Bus arrival time prediction using artificial neural network model. Proceedings. 7th Int. IEEE Conf. Intell. Transp. Syst. (IEEE Cat. No.04TH8749), IEEE; n.d., p. 988–93. doi:10.1109/ITSC.2004.1399041.

[49] Dharia A, Adeli H. Neural network model for rapid forecasting of freeway link travel time. Eng Appl Artif Intell 2003;16:607–13. doi:10.1016/j.engappai.2003.09.011.

[50] Yin T, Zhong G, Zhang J, He S, Ran B. A prediction model of bus arrival time at stops with multi-routes. Transp Res Procedia 2017;25:4623–36. doi:10.1016/j.trpro.2017.05.381.

[51] Vanajakshi L, Rilett LR. A comparison of the performance of artificial. neural networks and support vector machines for the prediction of traffic speed. IEEE Intell. Veh. Symp. 2004, IEEE; n.d., p. 194–9. doi:10.1109/IVS.2004.1336380.

[52] Zhong S., Hu J b., Ke S., Wang X., Zhao J., Yao B. A Hybrid Model based on Support Vector Machine for Bus Travel-Time Prediction. Promet - Traffic - Traffico 2015;27:291–300. doi:10.7307/ptt.v27i4.1577.

[53] Muller K-R, Mika S, Ratsch G, Tsuda K, Scholkopf B. An introduction to kernel-based learning algorithms. IEEE Trans Neural Networks 2001;12:181–201. doi:10.1109/72.914517.

[54] Bin Y, Zhongzhen Y, Baozhen Y. Bus Arrival Time Prediction Using Support Vector Machines. J Intell Transp Syst 2006;10:151–8. doi:10.1080/15472450600981009.

[55] Liu H, Van Zuylen H, Van Lint H, Salomons M. Predicting Urban Arterial Travel Time with State-Space Neural Networks and Kalman Filters. Transp Res Rec J Transp Res Board 2006;1968:99–108. doi:10.1177/0361198106196800112.

[56] van Lint JWC, Hoogendoorn SP, van Zuylen HJ. Accurate freeway travel time prediction with state-space neural networks under missing data. Transp Res Part C Emerg Technol 2005;13:347–69. doi:10.1016/j.trc.2005.03.001.

[57] Kisgyörgy L, Rilett LR. Travel time prediction by advanced neural network. Period Polytech Civ Eng 2002;46:15–32.

[58] Ishak S, Alecsandru C. Optimizing Traffic Prediction Performance of Neural Networks under Various Topological, Input, and Traffic Condition Settings. J Transp Eng 2004;130:452–65. doi:10.1061/(ASCE)0733-947X(2004)130:4(452).

[59] Sanghoon Bae, Kachroo P. Proactive travel time predictions under interrupted flow condition. Pacific Rim TransTech Conf. 1995 Veh. Navig. Inf. Syst. Conf. Proceedings. 6th Int. VNIS. A Ride into Futur., IEEE; n.d., p. 179–86. doi:10.1109/VNIS.1995.518836.

[60] Hall RW, Vyas N. Buses as a Traffic Probe: Demonstration Project. Transp Res Rec J Transp Res Board 2000;1731:96–103. doi:10.3141/1731-12.

[61] Tantiyanugulchai S, Bertini RL. Arterial performance measurement using transit buses as probe vehicles. Proc. 2003 IEEE Int. Conf. Intell. Transp. Syst., IEEE; n.d., p. 102–7. doi:10.1109/ITSC.2003.1251929.

[62] Bertini RL, Tantiyanugulchai S. Transit Buses as Traffic Probes: Use of Geolocation Data for Empirical Evaluation. Transp Res Rec J Transp Res Board 2004;1870:35–45. doi:10.3141/1870-05.

[63] Padmanaban RPS, Vanajakshi L, Subramanian SC. Automated Delay Identification for Bus Travel Time Prediction towards APTS Applications. 2009 Second Int. Conf. Emerg. Trends Eng. Technol., IEEE; 2009, p. 564–9. doi:10.1109/ICETET.2009.43.

[64] Vasantha Kumar S, Vanajakshi L. Urban Arterial Travel Time Estimation Using Buses as Probes. Arab J Sci Eng 2014;39:7555–67. doi:10.1007/s13369-014-1332-z.

[65] Shalaby A, Farhan A. Bus Travel Time Prediction Model for Dynamic Operations Control and Passenger Information Systems. Transp. Res. Board 82nd Annu. Meet., 2003.

[66] El Esawey M, Sayed T. A framework for neighbour links travel time estimation in an urban network. Transp Plan Technol 2012;35:281–301. doi:10.1080/03081060.2012.671028.

[67] Bae S. Dynamic estimation of travel time on arterial roads by using automatic vehicle location (AVL) bus as a vehicle probe 1995.

[68] Chakroborty P, Kikuchi S. Using bus travel time data to estimate travel times on urban corridors. Transp Res Rec J Transp Res Board 2004:18–25.

[69] Jeong R, Rilett LR. Prediction Model of Bus Arrival Time for Real-Time Applications. Transp Res Rec J Transp Res Board 2005;1927:195–204. doi:10.1177/0361198105192700123.

[70] Padmanaban RPS, Divakar K, Vanajakshi L, Subramanian SC. Development of a real-time bus arrival prediction system for Indian traffic conditions. IET Intell Transp Syst 2010;4:189. doi:10.1049/iet-its.2009.0079.

[71] Kieu LM, Bhaskar A, Chung E. Bus and car travel time on urban networks: integrating bluetooth and bus vehicle identification data 2012.

[72] Zhan X, Hasan S, Ukkusuri S V., Kamga C. Urban link travel time estimation using large-scale taxi data with partial information. Transp Res Part C Emerg Technol 2013;33:37–49. doi:10.1016/j.trc.2013.04.001.

[73] Arhin S, Stinson RZ. Transit Bus Travel Time Prediction using AVL Data. Int J Eng Res Technol 2016;5:21–6.

[74] Sharmila RB, Velaga NR, Kumar A. SVM-based hybrid approach for corridor-level travel-time estimation. IET Intell Transp Syst 2019;13:1429–39. doi:10.1049/iet-its.2018.5069.

[75] Mattera D, Haykin S. Support vector machines for dynamic reconstruction of a chaotic system. Adv. kernel methods Support vector Learn., 1999, p. 211–41.

[76] Reddy KK, Kumar BA, Vanajakshi L. Bus travel time prediction under high variability conditions. Curr Sci 2016:700–11.

[77] Pawar DS, Patil GR, Chandrasekharan A, Upadhyaya S. Classification of Gaps at Uncontrolled Intersections and Midblock Crossings Using Support Vector Machines. Transp Res Rec J Transp Res Board 2015;2515:26–33. doi:10.3141/2515-04.

[78] Vapnik VN. Computational Learning Theory. 1998.

[79] Burges CJC. A tutorial on support vector machines for pattern recognition. Data Min Knowl Discov 1998;2:121–67.

[80] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14:199–222. doi:10.1023/B:STCO.0000035301.49549.88.

[81] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20:273–97. doi:10.1007/BF00994018.

[82] Nagalla R, Pothuganti P, Pawar DS. Analyzing Gap Acceptance Behavior at Unsignalized Intersections Using Support Vector Machines, Decision Tree and Random Forests. ANT/SEIT, 2017, p. 474–81.

[83] Kenneth DL, Ronald KK. Advances in business and management forecasting. Bingley, UK Emerald Books 1982.