Soft computing-based traffic density estimation using automated traffic sensor data under Indian conditions

Jithin Raj, Hareesh Bahuleyan, V. Ramesh and Lelitha Devi Vanajakshi*

Department of Civil Engineering, Indian Institute of Technology Madras, Chennai 600 036, India

Traffic density is an indicator of congestion and the present study explores the use of data-driven techniques for real time estimation and prediction of traffic density. Data-driven techniques require large database, which can be achieved only with the help of automated sensors. However, the available automated sensors developed for western traffic may not work for heterogeneous and lane-less traffic. Hence, the performance of available automated sensors was evaluated first to identify the best inputs to be used for the chosen application. Using the selected data, implementation was carried out and the results obtained were promising, indicating the possibility of using the proposed methodology for real time traveller information under such traffic conditions.

Keywords: Automated traffic sensors, artificial neural network, *k*-nearest neighbour, traffic density.

WITH the fast growing urban population and increasing vehicle population, it is becoming difficult to implement efficient traffic management for Indian roads. Intelligent transportation systems (ITS) is viewed as an option to handle some of these issues and is becoming more popular under this scenario. ITS enable gathering of data and providing timely feedback to traffic managers and road-users based on the real time data. Advanced traveller information system (ATIS) is a major functional area of ITS and it deals with providing real time traffic information to travellers for making informed travel decisions. The information provided can include expected travel time, congestion condition, locations of incidents, weather and road conditions, optimal routes, recommended speeds and lane restrictions.

Traffic congestion information is most sought after, followed by travel time and travel speed information. However, congestion being qualitative in nature, there is a need to identify the best measure to quantify it. As per the Highway Capacity Manual (HCM)[1], traffic density on freeways, delays at signalized intersections and walking speed for pedestrians are examples of measures of effectiveness that characterize traffic conditions on a facility. Of these, traffic density is the primary measure for quantifying congestion on roadways, other than the intersection areas. Traffic density is the number of vehicles occupying a given length of roadway. Density being a spatial variable makes it difficult to carry out direct measurements. Aerial photography is the primary approach to measure density directly from field, which is very difficult to implement. Hence, it is usually estimated from other location-based parameters such as speed, flow or occupancy, making it a challenging research problem. This problem of estimation of density from location based parameters such as volume and time mean speed (TMS) obtained using selected sensors is considered in this study. This becomes more challenging under Indian conditions, with heterogeneous and lane less-traffic, leading to high variability and randomness. In addition, these traffic conditions make automated data collection difficult, leading to limited amount of data being available for building models. Consequently, there are not many studies reported on traffic density estimation under Indian traffic conditions using data-driven techniques and therefore is selected as the demonstration application in the present study.

Road users, in general, are more interested to know what they can expect while making the trip, rather than the present scenario, making prediction to future time intervals an important task. Hence, in this study, the traffic density is first estimated using speed and volume data from sensors. The estimated density values are then predicted to future time intervals by identifying the evolution pattern. The density at a future time interval is predicted from the density of previous time intervals. This density value can be provided to road users in the form of sign boards or messages in real time, if the estimation-prediction models are assembled along with a real time data collection technique. Thus, the accuracy of the final information provided to users depends on the quality of input data, and the estimation and prediction techniques used, which are discussed below.

The basic requirement of any successful ITS implementation is good quality real time traffic data. Thus, accurate traffic sensors are one of the essential elements of ITS implementation. Loop detectors were the most widely used detection technology for automated traffic

*For correspondence. (e-mail: lelitha@iitm.ac.in)

data collection all over the world. The major disadvantage of loop detectors is the intrusive nature that leads to installation and maintenance problems, traffic disruption and removal of pavement structure. Later, several types of non-intrusive traffic sensors were introduced for vehicle sensing. Some of the popular technologies include video image processing, radio frequency identification, bluetooth, infra-red, etc. However, most of these traffic sensors were developed for lane-based and homogeneous traffic conditions. Accurate and automated real time data collection continues to be a major challenge under heterogeneous and no-lane disciplined traffic conditions. Hence, there is a need to evaluate the performance of available traffic sensors before using it for any ITS application. The present study thus evaluates available detection technologies under Indian traffic conditions, before using them for the end application. Four popularly used automated sensors are evaluated based on statistical analysis and data from sensors that are found to be performing well are used for density estimation.

Artificial neural networks (ANN) and *k*-nearest neighbour (*k*-NN), which are popularly used machine learning techniques, have been explored for density estimation and prediction in this study. In the density estimation model, speed and volume are given as inputs to the model for the corresponding density as output. In the second model, the time series of estimated density values of previous time intervals is used to determine the density in the next time step. Therefore the model will be able to predict traffic density for a time interval if provided with volume and speed of previous time interval. The data required for the model were collected using the best performing sensor identified from the evaluation of sensors.

Literature review

A review of literature in the area of estimation of traffic parameters like density showed the use of various techniques such as historic-data based methods, statistical methods, machine learning techniques and model based techniques as the most commonly used ones. Chang and Gazis[2] estimated density using Kalman filtering technique from flow data obtained by aerial photography. Coifman[3] used loop detector data to estimate density based on the conservation of vehicles. In another study, Nahi and Trivedi[4] estimated density and speed from the flow data obtained by aerial photography. Wang *et al.*[5] estimated traffic flow variables such as flow, space mean speed and density using online parameter model estimator and discussed the estimator's tracking ability and sensitivity to various initial conditions. Darwish and Abu Bakar[6] provide a literature review of traffic density estimation. They classify the techniques into categories – infrastructure-based and infrastructure-free and list out the advantages and disadvantages of each. Bhaskar *et al.*[7] developed a data fusion model that integrates data from two sources (loop detectors and bluetooth scanners) for travel time and density estimation in a seamless and reliable manner.

Studies reporting traffic parameter prediction mainly focussing on data driven approaches are discussed here. Zhang and Rice[8] used a varying coefficients linear model, which varies as a smooth function of departure time, with past instantaneous travel time to predict future travel time. The principal component analysis and nearest neighbour approach were investigated by Rice and Zwet[9] by combining the historical data and the instantaneous travel time data. Other methods such as linear regression and Kalman filtering are widely used in the area of prediction. Kwon *et al.*[10] estimated travel time on a freeway, using flow and occupancy data obtained from loop detectors and predicted to future time steps using linear regression. Kalman filter method was employed by Chen and Chein[11] to predict travel time using data obtained from probe vehicles. Fabrizi and Ragona[12] developed a pattern matching method of prediction, which tried to identify patterns in the past data that describe the actual traffic load and produces forecasts supposing that the trend would repeat. Padiath *et al.*[13] compared the performance of a historic technique, an ANN-based technique and a model based technique to predict traffic density under Indian traffic conditions. However, the study used limited data collected manually. Ozkurt and Camci[14] presented an automatic traffic density estimation and vehicle classification method using ANN and 94% accuracy was reported. Vanajakshi and Rilett[15] predicted traffic speed ranging from 2 min ahead up to an hour into future using ANN and support vector machines and reported that performance of ANN largely depended on the amount of data. Li and Chen[16] developed a multilayer perception neural network model for freeway travel time prediction that required inputs such as volume and speed from loop detectors, historic travel time, most recently available current travel time, rainfall, and information on occurrence of accidents. Drakopoulos and Abdulkader[17] studied the neural network training of heterogeneous data and proposed data pruning (removal of noisy data) and ordered-training (partitioning of data) as effective methods to deal with heterogeneous data.

k-NN has also been used extensively for similar applications. Cruz *et al.*[18] studied the problem of processing *k*-NN queries in road networks considering traffic conditions and the queries return the *k*-points of interest that could be reached in the minimum amount of time. Zhang *et al.*[19] presented a method for short-term urban expressway flow prediction system with accuracy over 90% using *k*-NN. Lin *et al.*[20] applied *k*-NN method to form the training dataset for local linear wavelet neural network instead of taking the whole historical dataset for training for short-term prediction of five minutes volume. Xiaoyu *et al.*[21] proposed a two-tier *k*-NN algorithm combined with the actual traffic flow to improve the processing speed and the accuracy of the algorithm. Zheng and Su[22] developed a *k*-NN–LSPC (*k*-NN–linearly sewing principle component) for prediction of traffic volume, which outperformed eight other algorithms. Hodge *et al.*[23] addressed the problem of short-term prediction of traffic flow through a scalable neural network-based *k*-NN predictor.

The above reviews show the use of various model-based and data-driven approaches for estimation and prediction of traffic parameters. However, under Indian conditions, automated sensors were not available and hence prediction problems using data-driven approaches such as ANN or *k*-NN were not attempted exhaustively. Based on the literature and based on the availability of a good automated database, two data-driven approaches namely ANN and *k*-NN approach were selected for estimation and prediction of traffic density in the present study. Another factor on which the accuracy of these applications depend, in addition to the choice of correct estimation and prediction tool, is the accuracy of input data. This is critical while using automated data, which are more prone to errors. This issue is more severe under Indian conditions because the performance of these automated sensors has not been evaluated under such traffic conditions in the past. The present study evaluated the performance of available sensors and the best performing ones were used for the end application. A literature review in this area was also carried out and is provided below.

Traffic sensors in general can be classified as location-based and spatial-based, depending on the placement of the sensor and the nature of data being collected. Location-based sensors are more popular due to the ease of installation without any participation requirement from vehicles and it can easily collect data of the entire traffic population crossing the sensor location. Location-based sensors can be intrusive or non-intrusive based on whether they need to be placed below the road surface or not. Inductive loop detectors, pneumatic road tubes, magnetometer and piezo-electric tubes are some of the detectors that come under the category of intrusive sensors. Video image processing, microwave radar, acoustic, magnetic and infrared detectors are the popular non-intrusive sensors. Based on the sensing principle used, location-based sensors can be classified as inductive loop, radar-based, image processing-based, infra-red-based and so on[24]. Evaluation of various traffic sensors and comparison of performance were reported by various agencies from western countries, mainly USA[25–30].

It can be observed that all the above studies were carried out where homogeneous and lane-disciplined traffic existed. The traffic in western countries is dominated by cars and trucks, but countries like India have to deal with a variety of vehicles with wide range of speed and physical properties using any space in the roadway without following lane discipline. So, all these sensors reported to be performing well under homogeneous and lane-disciplined traffic may not perform equally well in a heterogeneous and no-lane disciplined condition. Therefore, the objective of the present study is to perform a similar evaluation under Indian conditions. The choice of sensors for the evaluation was based on findings from earlier studies as well as popularity in terms of field installations around the world. As most studies used just one or two measures such as correlation coefficient and per cent error for quantifying and comparing the performance of sensors, a detailed evaluation of four sensors for three different parameters using five statistical measures was adopted in the present study. The evaluation was carried out by comparing the sensor reported values and ground truth values thereby finding the strength and weakness of each detector.

Methodology

The adopted methodology for the present study can be highlighted as:

- Selection of best sensor for density estimation from available sensors.
- Data collection using the selected sensor.
- Development of estimation–prediction models using data mining.
- Performance evaluation.

Selection of best sensor for density estimation from the available sensors

The basic requirement of any of the ITS applications is an automated traffic sensor for capturing real-time traffic data. The sensors which are proven to be performing well in homogeneous traffic may not perform well in heterogeneous traffic because of presence of multiple classes of vehicles moving in a no-lane disciplined manner. Hence, the sensors need to be evaluated before using the data for any ITS applications under Indian traffic conditions. The study will analyse two different variables namely volume and speed from these sensors, which are the primary input for estimation–prediction model. The ground truth data collected manually were compared with the sensor reported numbers for accuracy. The data from the sensor which performs well will be used in the end application. Four types of automated sensors namely Smartsensor HD[31], Traficam Collect-R[32], TRAZER[33] and TIRTL[34] were evaluated. All these sensors are location-based and provide classified vehicle count, speed, lane occupancy etc. Though these sensors are based on advanced technologies such as image processing, infra-red and radar and reported good performance under ideal conditions, the performance of these sensors in Indian traffic condition needs to be addressed and is attempted.

The performance of the selected sensors was evaluated by comparing the sensor reported values with the corresponding actual or ground truth values. Total count and speed were evaluated using the statistical measures for each one-minute interval. Each of the statistical measures has advantages and disadvantages and a single measure may not be able to represent the performance completely. Hence, the evaluations in this study used five different statistical measures namely mean absolute error (MAE)[35,36], mean absolute percentage error (MAPE)[37,38], correlation coefficient[39], GEH statistic[40], and Theil's inequality coefficient, U[41]. U can be split into three components U_m, U_s and U_c which indicate the nature of difference between two sets of data. U_m is the bias proportion, which indicates the proportion of the inequality contributed to a systematic tendency towards wrong (over or under counting) estimation of the true value. U_s is the variance proportion, which indicates the proportion of the inequality attributed to unequal variances between the detector's reported values and true values. U_c is the covariance proportion, which indicates the proportion of the inequality that is unsystematic. Sample results obtained are discussed below.

Volume analyses: Volume, which is defined as the number of vehicles passing a point during a given time interval, is analysed per minute in this study. Ground truth values were measured manually from videos. The peak and off-peak traffic conditions were analysed separately. The peak hours were considered as 8–11 am and off-peak hours were considered as 11 am–3 pm. Since TIRTL can give per vehicle data, analysis was also conducted on individual identification of vehicles. Each vehicle reported by the sensor was manually identified in the video and the number of missed and false identifications was found out. Table 1 shows the results obtained for volume analysis.

Speed analyses: Analyses similar to volume were carried out for speed values. Ground truth speed values were collected using laser gun[42] and the values were averaged over one minute interval for comparing with the corresponding data given by the sensors (except TIRTL). Analysis period considered was around 30 min. To evaluate TIRTL, the individual speed of each vehicle identified by the sensor was compared with speed measured in the field using laser gun for the corresponding vehicle. This involved matching of individual vehicles' speeds. Table 1 shows the results obtained for speed analysis with the best fit values for each of these measures.

For volume evaluation, it can be seen that, in terms of both MAE and MAPE, Trazer and TIRTL have less error and Smartsensor and Traficam showed bigger errors. Considering correlation coefficient r, Smartsensor, TIRTL and Trazer have linear relationship between actual and sensor reported values, with Trazer having values more close to 1 showing stronger relations. Traficam showed a wide range of values between –0.44 and 0.86, indicating linear inverse relationship between sensor and ground truth values in certain cases. In GEH statistic, TIRTL has values more close to zero showing less error. Others have values in the range of 1 and 4. The U values of Trazer and TIRTL are less than 0.1, indicating better performance. U_c values are also reasonable for both these devices. Even though the error in the other two sensors is similar, U_c is higher in Traficam, showing the best proportioning of error with U_c values close to 1. The systematic error in Smartsensor is higher in peak conditions, which can be interpreted as sensor not being able to perform well with increased number of vehicles. Overall, it can be concluded that TRAZER and TIRTL perform better in terms of volume out of the four sensors analysed. However, when TIRTL was tested under non-ideal conditions such as near an intersection, the accuracy reduced. The portable TIRTL near an intersection was able to get total volume with accuracy of 90–95% in an hour, whereas under free flow conditions it was above 98%.

For speed evaluation, it can be seen that TIRTL and Smartsensor follow closely with ground truth values. The other two sensors, Traficam and Trazer had a wide range of values between 10 and 30 in case of MAE and 15 to 50 in case of MAPE. None of the sensors showed any linear positive relations permanently, as the results were mixed in nature. The range of speed values was small, as the considered time interval was only 30 min and hence correlation could not be considered as a good statistical measure for evaluating speed within such a small range. The GEH statistic showed better result for Smartsensor and TIRTL compared to the other two sensors. The Theil's inequality coefficient, U also showed better result for Smartsensor and TIRTL compared to other two sensors. Trazer and Traficam had U values ranging between 0.1 and 0.7, whereas the other two had values less than 1 and 0.1 respectively, with co-variance proportion U_c closer to 1. Overall, it can be concluded that Smartsensor and TIRTL are better options for speed detection under Indian scenario. However, TIRTL, the infra-red sensor had better performance while considering volume as well as speed and was selected for the present study.

Data collection

The collection of data required for the estimation-prediction models was achieved with the help of best performing sensor among the available sensors. Volume, speed and traffic density data were required for training and validation of these models. Volume is the number of vehicles passing a point on a roadway in unit time. Time mean speed (TMS) is defined as the arithmetic average of individual vehicle speeds passing a point on a roadway.

Table 1. Evaluation of different sensors for volume and speed

	MAE	MAPE	r	GEH	U	U_m	U_s	U_c
Best fit	0	0	1	0	0	0	0	1
Volume (peak condition)								
Smartsensor	07.45–26.86	24.80–30.63	0.54–0.93	1.45–3.09	0.16–0.27	0.10–0.56	0.00–0.43	0.08–0.90
Trazer	02.19–07.40	06.04–13.55	0.72–0.99	0.37–1.04	0.04–0.09	0.01–0.73	0.00–0.16	0.22–0.82
Traficam	09.90–33.67	14.47–53.88	–0.44–0.86	1.19–4.02	0.08–0.27	0.03–0.11	0.00–0.05	0.86–0.96
TIRTL	01.53–02.68	02.42–04.87	0.98–0.99	0.198–0.353	0.02–0.03	0.17–0.45	0.01–0.09	0.54–0.75
Volume (off-peak condition)								
Smartsensor	06.77–14.53	17.17–37.02	0.40–0.90	1.06–2.30	0.10–0.22	0.21–0.59	0.08–0.31	0.30–0.63
Trazer	02.44–09.00	04.63–14.37	0.76–0.98	0.41–1.19	0.03–0.09	0.02–0.71	0.00–0.24	0.30–0.81
Traficam	12.97–20.00	25.65–43.32	–0.64–0.24	1.15–2.75	0.15–0.22	0.00–0.07	0.00–0.01	0.92–0.98
TIRTL	01.40–01.43	02.29–02.42	0.98–0.98	0.181–0.186	0.02–0.02	0.15–0.44	0.02–0.09	0.54–0.75
Speed								
Smartsensor	03.07–05.19	05.57–09.63	–0.21–0.83	0.42–0.72	0.04–0.06	0.00–0.46	0.00–0.17	0.52–0.95
Trazer	08.43–20.32	19.19–42.01	–0.35–0.69	1.37–3.41	0.14–0.74	0.57–0.79	0.01–0.15	0.14–0.41
Traficam	08.50–28.15	16.46–51.89	–0.40–0.31	1.18–4.46	0.10–0.67	0.04–0.94	0.00–0.16	0.05–0.80
TIRTL	01.10–01.27	02.94–04.14	0.98–0.99	0.18–0.22	0.01–0.02	0.00–0.05	0.00–0.01	0.85–0.99

Traffic density is the number of vehicles occupying a given length of roadway. Of these, volume and speed can be obtained from location-based sensors and traffic density has to be obtained indirectly for validating the results.

For performance evaluation, sensors were installed along Rajiv Gandhi Salai, a three lane urban road in Chennai, India. This road is a representative Indian road with heterogeneous and lane-less traffic conditions, i.e. different class of vehicles like two-wheelers, three-wheelers, passenger cars, bus and trucks use the road space without any segregation. Smartsensor HD and Traficam Collect-R were installed at the second foot-over bridge near Indira Nagar railway station facing southbound and northbound traffic respectively. The videos for Trazer were collected using IP camera located at third foot-over bridge near Thiruvanmiyur railway station, at a distance of 730 m from the previous location, facing northbound traffic. TIRTL was installed near Perungudi toll gate, 6 km from Trazer location facing southbound traffic. Figure 1 shows a Google map image highlighting these points.

Estimation and prediction of traffic density

The present study uses data mining approaches – *k*-NN and ANN for estimation and prediction of density from speed and volume obtained from automated sensor. For estimation of traffic density, the flow and speed were selected as inputs to get the corresponding density as the output. In order to predict density values to future time intervals, the previous several estimated density values were used as input. Figures 2 and 3 show the framework used for density estimation and prediction respectively. Machine learning-based data mining approaches used for estimation and prediction are detailed below.

Data mining approaches

The basis of data mining is a process of using tools to extract useful knowledge from large datasets[43]. Machine learning techniques namely, *k*-NN and ANN were selected as tools for data mining in this study, based on acceptable performance of the same reported in earlier studies.

k-Nearest neighbour algorithm[43]*:* *k*-NN is one of the simplest machine learning algorithms, most widely used for classification. It is a non-parametric and supervised algorithm, which classifies a new unclassified record by comparing it to 'similar' records in the training data set. The most common method of defining 'similarity' is based on the Euclidean distance between the records in the feature space. In this study, the *k*-NN regression method was utilized, where the output was not a pre-defined class, but a continuous value.

In the density estimation process carried out in this study, the *k*-NN algorithm was trained with flow and speed provided as inputs to predict the output variable, the density. This is explained with the help of Figure 4. The triangles in the figure are the training data points positioned in the two-dimensional feature space based on their flow and speed values. Each training data point has a corresponding density value. When a new test input is provided (indicated by solid circle), based on its speed and flow values, it occupies a position in the same feature space. Next, the algorithm calculates the distance between the new test data point and each of the training data points. It identifies the k nearest neighbours, in terms of Euclidean distance. For example, if $k = 3$, the output density for the new test record would be calculated as the average of densities of the three triangles shown within

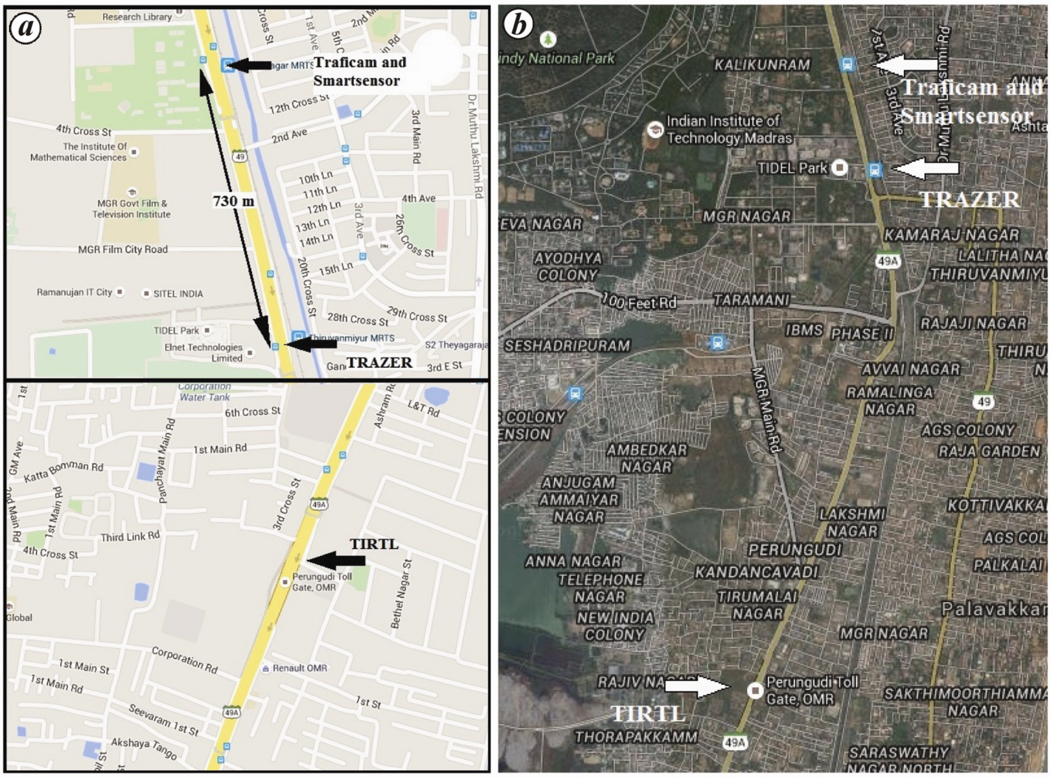

Figure 1. Location of sensors. ***a***, Map; ***b***, Satellite. (Source: Google maps.)

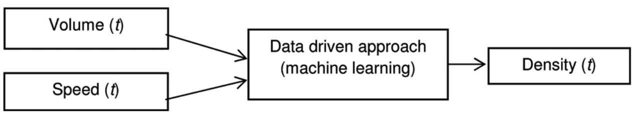

Figure 2. Framework for the estimation of traffic density.

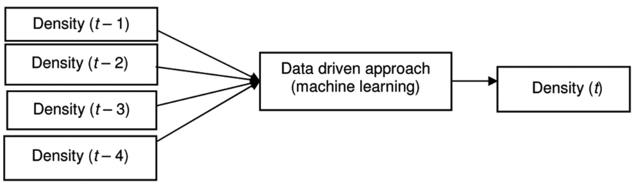

Figure 3. Framework for prediction of traffic density.

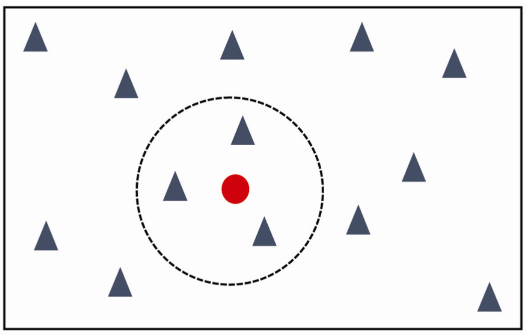

Figure 4. Demonstration of *k*-NN algorithm.

the dashed circle, as these are the three closest neighbours. Let *s1*, *s2* and *s3* be the speeds; *v1*, *v2* and *v3* the flows; and *d1*, *d2* and *d3* are the density of data points indicated by these triangles. For the test data point (red circle), let the speed be *s* and flow be *v*, which are provided as input and it is required to estimate the corresponding density *d*. Then the Euclidean distances are given by eq. (1):

$$P_1 = \sqrt{(s_1 - s)^2 + (v_1 - v)^2},$$

$$P_2 = \sqrt{(s_2 - s)^2 + (v_2 - v)^2},$$

$$P_3 = \sqrt{(s_3 - s)^2 + (v_3 - v)^2}. \tag{1}$$

Since triangles 1, 2 and 3 are the closest neighbours, any distance P_i, $i \neq 1$, 2 and 3, will only be greater than P_1, P_2 and P_3. The estimated density for the new record would be calculated as a simple arithmetic average as given in eq. (2). The number of nearest neighbours considered in this case was 3 (i.e. $k = 3$), based on a sensitivity analysis.

$$d = \frac{d_1 + d_2 + d_3}{3}. \tag{2}$$

In the second application of *k*-NN algorithm discussed in this article, the density of the current time step is to be predicted based on densities of the previous n time steps, which are provided as inputs to train the algorithm. This implies that the training set records would be positioned in an n-dimensional feature space. Here, the k-NN algorithm looks for training samples in the historic data that have similar previous n densities as that of the test record. The value of n was selected based on an entropy analysis and the value of k was identified in this case based on sensitivity analysis which was 10.

Artificial neural network: ANN is a popular machine learning tool inspired by biological nervous system and is composed of units operating in parallel. Neural networks can be trained to perform a particular function by adjusting the weights of the connections between units. Each of these processing units is known as neurons. Neural networks are trained to adjust the weights of these neurons so that a particular input leads to a specific target. In each neuron, scalar input p is transmitted through a connection that multiplies it by the scalar weight w and then added by a bias value b, to form the result $wp + b$. This sum is the input for the transfer function f which gives an output $f(wp + b)$. The basic idea is to get the best set of values for variables w and b in order to minimize the error between the actual output and the predicted value. The output of neurons in each layer will be the input to the neurons in the subsequent layers. The transfer function f can be hardlim, sigmoid, purelin, etc. based on the requirement. After passing through all the connected neurons, it should be able to produce the desired target.

There are generally four steps in ANN training using MATLAB and are explained below:

- Data assembling: Two sets of data are required for training – inputs and targets. The available data set was divided into two different subsets – training data set and testing data set. The main difference between these two data sets is that training data is used in training the neural network and the test data set is used for finding the performance and accuracy of the trained model.
- Network architecture and initialization of weights: Feed-forward networks often have one or more hidden layers of sigmoid neurons followed by an output layer of linear neurons. The number of hidden layers and neurons can be defined while creating the network. Weights are randomly initialized and as the training progresses, weights will change their magnitude to obtain the best relation.
- Training the network: Training was done using the inbuilt function in MATLAB. This function uses back-propagation algorithm, which automatically recalculates and minimizes the error at the end of every iteration. Minimization is done by updating the network weights and biases in the direction in which the performance function decreases most rapidly. Training continues till one of the stopping criteria is reached, which may be the maximum number of validation checks, number of iterations, or performance in terms of mean squared error.
- Simulation: Another set of data set (testing data) is provided to the trained network for predicting the corresponding output. The outputs can be compared with actual data for finding the performance of network.

Combining kNN with ANN: Along with testing ANN and k-NN individually, a model which is a fusion of two techniques was also attempted. The fusion methodology adopted is as follows: The volume and speed are provided as inputs to the k-NN algorithm. k-NN is required to determine the first k (taken as 100 in this study) nearest neighbours of the new input record, which in turn would form the training dataset for the ANN. Once ANN is trained, it can now predict a value of density for the original record. The flow chart of the fusion model is shown in Figure 5.

Implementation and results

The performance evaluation of sensors showed that infrared-based sensor, TIRTL outperformed all the other sensors in terms of volume and speed. Hence, the data required for the present application were collected using this sensor. Volume, speed and occupancy were collected and densities were calculated from the occupancy using the standard occupancy-density relation[44]

$$k = \frac{\%\text{occ}}{L_v + L_d}, \tag{3}$$

where %occ is the percentage of time detector being occupied by vehicles, L_v the vehicle length, and L_d is the detector length. In this study, L_v was obtained from the infrared sensor, which provided axle-to-axle length of every vehicle identified and L_d is the standard sensor length of the infrared sensor, which was 0.15 m.

The sensor works continuously for 24 h all days and stores data. One week's data was used for training the machine learning algorithms and the next week's data was used for validation. Figures 6 and 7 show the cumulative flows and average speed aggregated at 5 min intervals for a sample day.

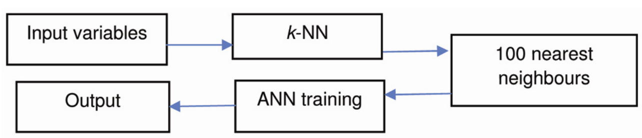

Figure 5. Model for combining ANN with k-NN.

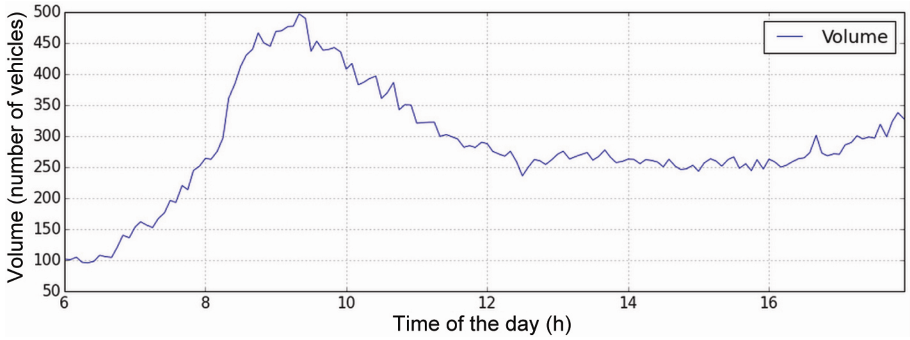

Figure 6. Flow (at 5 min intervals) for different hours of the day.

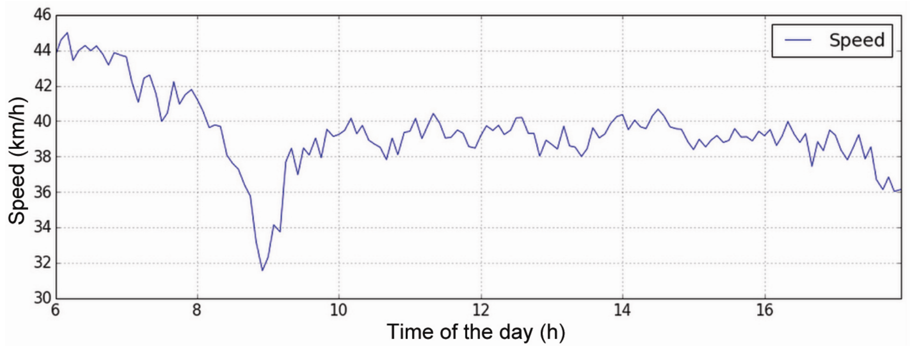

Figure 7. Speed (averaged at 5 min intervals) for different hours of the day.

The implementation of ANN was carried out in MATLAB using back-propagation algorithm for training[45]. For *k*-NN, the algorithm was implemented using an open source programing language and software environment for statistical computing, namely *R*[46]. The package for fast nearest neighbour (FNN) search algorithm in *R* was utilized[47]. To evaluate the performance of the algorithm, the results obtained were compared with the actual value of the output variable. The error was quantified using MAPE.

Estimation of traffic density

The available data set were divided into two subsets – training data set and testing data set. Four days data were taken as the training data set, and testing data set was selected as the same days of next week. The training set had speed and flow data at every 5 min interval as input and the corresponding actual density, obtained from the occupancy values, as the target variable. Once the network was trained, testing was carried out. Figure 8 shows the comparison between actual density and density estimated by each of the algorithms for a sample day.

From Figure 8, it can be observed that the estimated value of density is in line with actual density. MAPE values obtained are shown in Figure 9 for the four testing days. Though all the methods gave MAPE in the range of 1.5–4.5%, it can be observed that the performance of the ANN algorithm is comparatively better. The results of the combined ANN–*k*-NN model do not show any improvement in performance. This may indicate that giving larger training data set is more important for ANN's performance than giving a reduced set of best inputs.

Analysis was also carried out to check the performance of the models during peak and off-peak periods of the day separately and is shown in Figure 10. It can be observed from this figure that the *k*-NN model performed poorly during peak hours when compared to its performance during off-peak hours. The ANN model and the combined model, on the other hand, gave equally good performance even during peak hours.

Prediction of traffic density

The available data set were divided into two subsets – training data set and testing data set. Four days data were taken as the training data set, and testing data set was selected as the same days of next week. The training set had traffic density data of previous time intervals at every 5 min interval as input and the corresponding actual density as the target variable. To find out the optimum number of previous densities, *n*, the approximate entropy (ApEn) technique was used. ApEn is a technique used to quantify the amount of regularity and unpredictability of fluctuations in data over time[48]. ApEn obtained for the training data with respect to the number of previous densities is shown in Figure 11. It can be observed that the uncertainty in prediction is negligible if the densities of the previous four or more intervals were used as input.

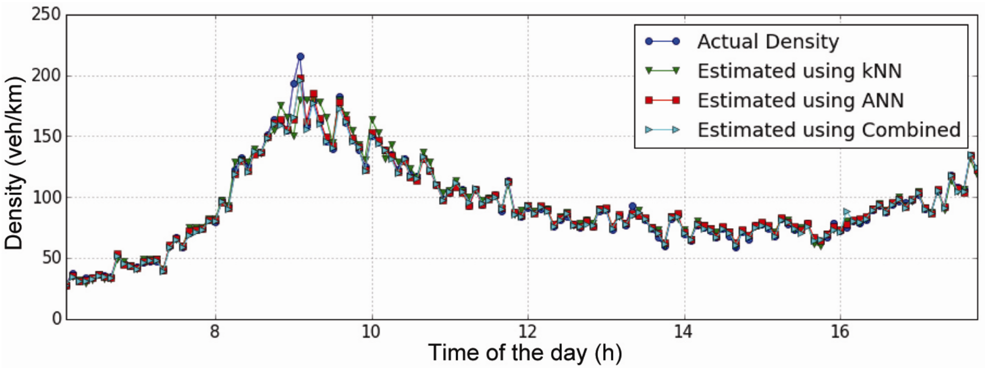

Figure 8. Sample comparison of actual and estimated density.

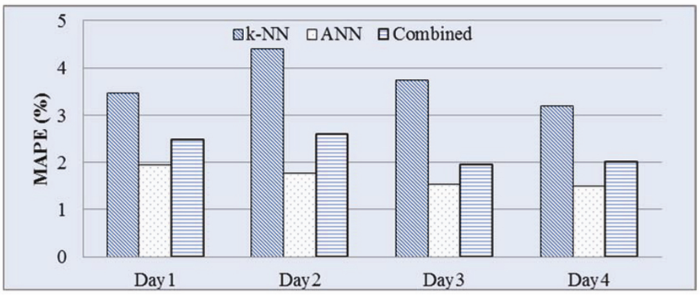

Figure 9. MAPE for density estimation.

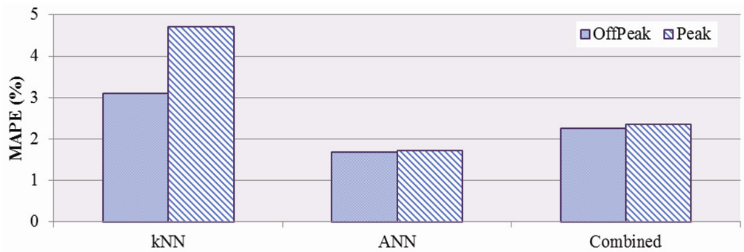

Figure 10. MAPE for peak versus off-peak periods for density estimation.

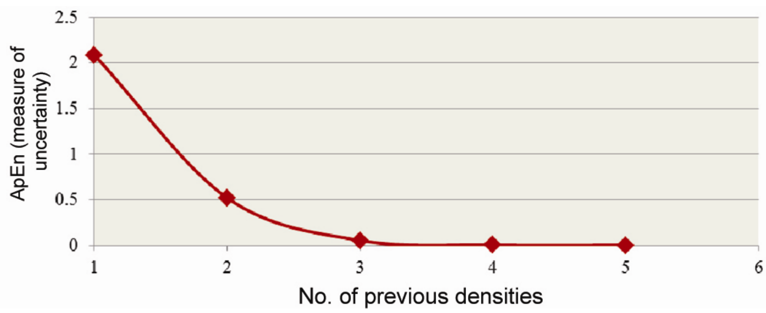

Figure 11. ApEn versus number of previous densities.

Hence, in this study, previous four time intervals density are taken as input (which is equivalent to previous 20 min) to predict the density in the next time intervals. The scheme was implemented and a sample result obtained is shown in Figure 12.

It can be observed that all the methods are able to capture the trend in the actual data. Errors were quantified for all test days and a bar plot of MAPE is shown in Figure 13. It can be seen that the performance of all the methods is comparable in this case. MAPE is in the range of

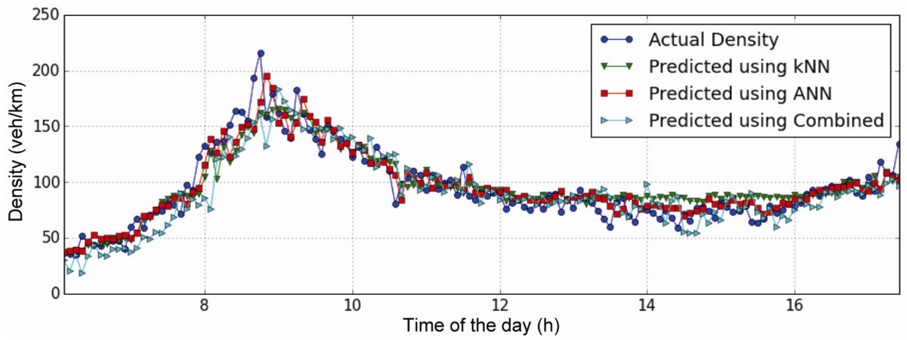

Figure 12. Sample comparison of actual and predicted density.

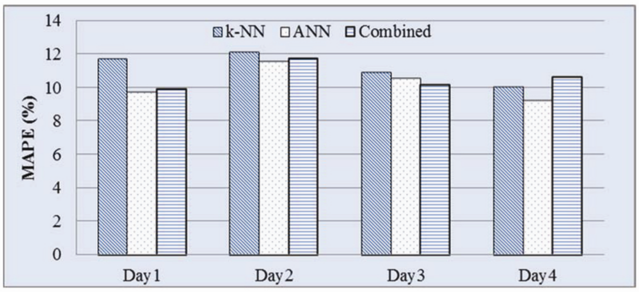

Figure 13. Bar plot of MAPE for density prediction.

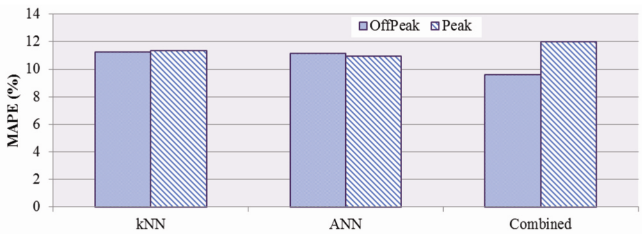

Figure 14. MAPE for peak versus off-peak periods for density prediction.

10–12% for all test cases, with ANN showing slightly better performance than the other two. Comparison of performance during peak and off-peak periods separately was also carried out and the results obtained are shown in Figure 14. From the results, it can be seen that the performance of *k*-NN and ANN is comparable with no significant difference in MAPE for the two periods. The combined model performed better during the off-peak periods. However, during peak periods, the accuracy of the combined model is seen to be lower.

Conclusions

The present study first evaluated available automated traffic sensors to identify the input data set to be used for the estimation and prediction problem. Four different sensors namely, Smartsensor, Trazer, Traficam Collect-R and TIRTL were compared based on one-minute interval analysis of traffic volume, and average speed. TIRTL outperformed the other sensors in terms of volume and speed and was used for the estimation–prediction process.

Two machine learning techniques namely ANN and *k*-NN were used in model development. A fusion model, which used the output from *k*-NN as a training set for ANN was also evaluated. The density estimation model used speed and volume as inputs. This produced MAPE in the range of about 1.5–4.5% using 5 min interval data. In the density prediction model, estimated density from previous time steps was used to predict future values and the MAPE obtained was in the range of 10–12%. The performance of all three approaches was comparable, with ANN showing a slight advantage over the other two. Combining *k*-NN and ANN did not show any significant improvement in performance. This may be due to reduction in training set in the combined approach, indicating that for better performance of ANN, more training data is important than providing significant input.

1. Highway Capacity Manual, Transportation Research Board. *National Research Council*, Washington, DC, 2000, vol. 113.
2. Chang, M. F. and Gazis, D. C., Traffic density estimation with consideration of lane-changing. *Trans. Sci.*, 1975, **9**(4), 308–320.
3. Coifman, B., Estimating density and lane inflow on a freeway segment. *Trans. Res. Part A: Policy Pract.*, 2003, **37**(8), 689–701.
4. Nahi, N. E. and Trivedi, A. N., Recursive estimation of traffic variables: section density and average speed. *Trans. Sci.*, 1973, **7**(3), 269–286.
5. Wang, Y., Papageorgiou, M. and Messmer, A., Real-time freeway traffic state estimation based on extended Kalman filter: a case study. *Trans. Sci.*, 2007, **41**(2), 167–181.
6. Darwish, T. and Bakar, K. A., Traffic density estimation in vehicular ad hoc networks: a review. *Ad Hoc Networks*, 2015, **24**, 337–351.
7. Bhaskar, A., Tsubota, T. and Chung, E., Urban traffic state estimation: fusing point and zone based data. *Trans. Res. Part C: Emerging Technol.*, 2014, **48**, 120–142.
8. Zhang, X. and Rice, J. A., Short-term travel time prediction. *Trans. Res. Part C: Emerg. Technol.*, 2003, **11**(3), 187–210.
9. Rice, J. and Zwet, E., A simple and effective method for predicting travel times on freeways. *IEEE Trans. Intell. Transp. Syst.*, 2004, **5**(3), 200–207.
10. Kwon, J., Coifman, B. and Bickel, P., Day-to-day travel time trends and travel time prediction from loop detector data. *Transp. Res. Rec.*, 2002, **17**(17), 120–129.
11. Chen, M. and Chien, S., Dynamic freeway travel time prediction using probe vehicle data: link based versus path based. *Transp. Rese. Rec.: J. Trans. Res. Board*, 2001, **1768**, 157–161.

12. Fabrizi, V. and Ragona, R., A pattern matching approach to speed forecasting of traffic networks. *Eur. Trans. Res. Rev.*, 2014, **6**(3), 333–342.
13. Padiath, A., Vanajakshi, L., Subramanian, S. C. and Manda, H., Prediction of traffic density for congestion analysis under Indian traffic conditions. In Intelligent Transportation Systems, ITSC'09. 12th International IEEE Conference on IEEE, 2009, pp. 1–6.
14. Ozkurt, C. and Camci, F., Automatic traffic density estimation and vehicle classification for traffic surveillance systems using neural networks. *Math. Comput. Appl.*, 2009, **14**(3), 187–196.
15. Vanajakshi, L. and Rilett, L. R., A comparison of the performance of artificial neural networks and support vector machines for the prediction of vehicle speed. In IEEE Intelligent Vehicles Symposium, Parma, Italy, 2004, pp. 194–199.
16. Li, C. S. and Chen, M. C., Identifying important variables for predicting travel time of freeways with non-recurrent congestion with neural networks. *Neural Comput. Appl.*, 2013, **23**(6), 1611–1629.
17. Darkopoulos, J. A. and Abdulkader, A., Training neural networks with heterogeneous data. *Neural Net.*, 2005, **18**, 595–601.
18. Cruz, L. A., Nascimento, M. A. and Macedo, J. A. F., *k*-Nearest neighbours queries in time-dependent road networks. *J. Inf. Data Manage.*, 2012, **3**(3), 211–226.
19. Zhang, L., Liu, Q., Yang, W., Wei, N. and Dong, D., An improved *k*-nearest neighbour model for short-term traffic flow prediction. *Proc. Soc. Behav. Sci.*, 2013, **96**, 653–662.
20. Lin, L., Li, Y. and Sadek, A., A *k*-nearest neighbour-based local linear wavelet neural network model for on-line short-term traffic volume prediction. *Proc. Soc. Behav. Sci.*, 2013, **96**, 2066–2077.
21. Xiaoyu, H., Yisheng, W. and Siyu, H., Short-term traffic flow forecasting-based on two-tier *k*-nearest neighbour algorithm. *Proc. Soc. Behav. Sci.*, 2013, **96**, 2529–2536.
22. Zheng, Z. and Su, D., Short-term traffic volume forecasting: a *k*-nearest neighbour approach enhanced by constrained linearly sewing principle component algorithm. *Transp. Res. Part C: Emerg. Technol.*, 2014, 143–157; 10.1016/j.trc.2014.02.009.
23. Hodge, V. J., Krishnan, R., Austin, J., Polak, J. and Jackson, T., Short-term prediction of traffic flow using a binary neural network. *Neural Comput. Appl.*, 2014, **25**(7–8), 1639–1655.
24. Klein, L. A., Mills, M. K. and Gibson, D. R., *Traffic Detector Handbook: Volume II* (No. FHWA-HRT-06-139), 2006.
25. Minge, E., Kotzenmacher, J. and Peterson, S., Evaluation of non-intrusive technologies for traffic detection (No. MN/RC 2010-36). Minnesota Department of Transportation, Research Services Section, 2010.
26. Grone, B. W., An evaluation of non-intrusive traffic detectors at the NTC/NDOR detector test bed, M Sc Thesis, The Graduate College at the University of Nebraska, 2012.
27. Middleton, D. and Parker, R., Vehicle detector evaluation (No. FHWA/TX-03/2119-1). Texas Transportation Institute, A&M University System, Texas, 2002.
28. Kotzenmacher, J., Minge, E. D. and Hao, B., Evaluation of portable non-intrusive traffic detection system (No. MN-RC-2005–37). Minnesota Department of Transportation, Research Services Section, 2005.
29. Middleton, D. R., Parker, R. and Longmire, R., Investigation of vehicle detector performance and ATMS interface (No. FHWA/TX-07/0-4750-2). Texas Transportation Institute, A&M University System, Texas, 2007.
30. Martin, P. T., Feng, Y. and Wang, X., Detector technology evaluation (No. MPC Report No. 03-154). Mountain-Plains Consortium, 2003.
31. SmartSensor, H. D., Technical Specification, Wavetronix, USA, 2012.
32. Traficam Collect-R User Guide, Traficon, Belgium, 2010.
33. Trazer Online User Manual, Kritikal Solutions Private Limited, India, 2010.
34. TIRTL Technical Overview, Version 2.1, CEOS Industrial, Heidelberg, Australia, 2010.
35. Wun, L. and Pearn, W. L., Assessing statistical characteristics of the mean absolute error for forecasting. *Int. J. Forecast.*, 1991, **7**, 335–337.
36. Willmott, C. J. and Matsuura, K., Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Res.*, 2005, **30**(1), 79.
37. Makridakis, S. and Hibon, M., The M3-competition: results, conclusions and implications. *Int. J. Forecast.*, 2000, **16**(4), 451–476.
38. McKenzie, J., Mean absolute percentage error and bias in economic forecasting. *Econ. Lett.*, 2011, **113**(3), 259–262.
39. Taylor, R., Interpretation of the correlation coefficient: a basic review. *J. Diag. Med. Sonography*., 1990, **6**(1), 35–39.
40. Traffic analysis toolbox, US Department of Transportation, Federal Highway Administration; http://ops.fhwa.dot.gov/traffic-analysistools/tat_vol3/sect5.htm (accessed on 26 June 2014).
41. Makridakis, S. and Hibbon, M., Evaluating accuracy (or error) measures, INSEAD working paper, 95/18/TM, 1995.
42. TruCAM User's Manual, Laser Technology, Inc, 2009, 2nd edn.
43. Larose, D. T., K-nearest neighbor algorithm. In *Discovering Knowledge in Data: An Introduction to Data Mining*, John Wiley, New York, 2005, pp. 90–106.
44. May, A. D., *Fundamentals of Traffic Flow*, Prentice–Hall, Inc. Englewood Cliff, New Jersey, 1990, 2nd edn.
45. Demuth, H. and Beale, M., *Neural Network Toolbox for Use with MATLAB*, 1993.
46. R Core Team, R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Australia, 2013.
47. Beygelzimer, A., Kakadet, S., Langford, J., Arya, S., Mount, D. and Li, S., *Fast Nearest Neighbor Search Algorithms and Applications*, Package FNN version 1.1, Technical report, R CRAN, 2013; http://cran.r-project.org/web/packages/FNN/FNN.pdf (accessed on 16 October 2014).
48. Pincus, S.M., Approximate entropy as a measure of system complexity. *Proc. Natl. Acad. Sci.*, 1991, **88**(6), 2297–2301.

ACKNOWLEDGEMENTS. The data collection effort in this project was made possible through sub-project CIE/10-11/169/IITM/LELI under the Centre of Excellence in Urban Transport project funded by the Ministry of Urban Development, Government of India, through letter No. N-11025/30/2008-UCD. The authors also acknowledge the technical support from all the sensor providers.

Received 5 July 2015; revised accepted 11 September 2016

doi: 10.18520/cs/v112/i05/954-964