Comparison of Model Based and Machine Learning Approaches for Bus Arrival Time Prediction

Vivek Kumar
Under Graduate student
Department of Civil Engineering,
Indian Institute of Technology Kharagpur
Kharagpur 721 302
INDIA
E-mail: viks.sai@gmail.com

B. Anil Kumar
Graduate student
Department of Civil Engineering,
Indian Institute of Technology Madras
Chennai 600 306
INDIA
E-mail: raghava547@gmail.com

Lelitha Vanajakshi
Associate Professor
Department of Civil Engineering
Indian Institute of Technology Madras
Chennai 600 036
INDIA
Ph: 91 44 2257 4291, Fax: 91 44 2257 4252
E-mail: lelitha@iitm.ac.in

Shankar C. Subramanian
Associate Professor
Department of Engineering Design
Indian Institute of Technology Madras
Chennai 600 036
INDIA
Ph: 91 44 2257 4705
E-mail: shankarram@iitm.ac.in

Paper Submitted for presentation and publication in Transportation Research Record, Transportation Research Board, National Research Council, Washington, D. C.

Word Count: Body text (3946) + Figures (1500) + Tables (750) = 6196.
Submitted on: August 1, 2013

1 Corresponding author
Comparison of Model Based and Machine Learning Approaches for Bus Arrival Time Prediction

ABSTRACT

The provision of accurate bus arrival information is critical to encourage more people to use public transport and alleviate traffic congestion. Developing a prediction scheme for bus travel times can provide such information. Prediction schemes can be data driven or may use a mathematical model that is usually less data intensive. This paper compares the performance of two methods – one being the data driven Artificial Neural Network (ANN) method and the other being the model based Kalman filter method, with regards to predicting bus travel time. The performances of both methods are evaluated using data collected from the field. It was found that the ANN based method performed slightly better in the presence of a large database but the Kalman filter method will be more advantageous when such a database is not available.
Comparison of Model Based and Machine Learning Approaches for Bus Arrival Time Prediction

INTRODUCTION

In recent times, traffic congestion has been increasing in Indian cities due to rapid changes in urbanization, economy levels, vehicle ownership and population growth that has lead to several negative impacts such as delays, pollution, etc. To overcome these problems, there is a need to provide more facilities by infrastructure expansion such as Bus Rapid Transit Systems (BRTS) (Ahmedabad), and Metro Rail Systems (Delhi, Hyderabad, Chennai, Bangalore). However, infrastructure expansion alone cannot meet the vehicular growth, and hence, there is a need to explore better traffic operations and management systems. One of the major challenges of traffic management in most of the developing countries such as India is due to heterogeneous traffic, which comprises both motorized and non-motorized vehicles with diverse vehicular characteristics. The motorized or fast moving vehicles include passenger cars, buses, trucks, auto rickshaws and motor cycles, whereas the non-motorized or slow moving vehicles include bicycles, cycle rickshaws and animal drawn carts. The use of Intelligent Transportation Systems (ITS) for operation and management of traffic is a better option that is gaining interest in recent years. Attracting more travelers towards public transportation system is one way to reduce congestion, which comes under Advanced Public Transportation Systems (APTS). APTS is one of the functional areas of ITS that can help to attract more people towards public transport. Predicting accurate bus travel times and providing reliable information to passengers is a popular APTS application. However, the information provided to passengers should be reliable; otherwise customers may reject the system due to lack of reliability (1). The reliability of such information being provided depends on the prediction technique and the input data used for the same.

There are many studies on prediction of travel time using various techniques such as historical and real-time approaches, statistical techniques, machine learning techniques and model-based techniques. However, there are only limited studies under heterogeneous traffic conditions such as the one existing in India. Machine learning techniques such as Artificial Neural Networks (ANN) and Support Vector Machines (SVM) are commonly used to predict travel time because of their ability to solve complex non-linear relationships. ANN has proved to be one of the most effective tools for pattern recognition across different sets of problems. Considering anomalies in datasets, which are true for travel time across a particular stretch of road, ANN seems to be a suitable candidate for prediction. This study attempts the short term Bus Travel Time Prediction (BTTP) by developing a neural network model taking most correlated previous trips of same day and same week in to account. In this study, a particular section of the road is divided into different subsections and the network is trained separately for each subsection. The disadvantage is that these types of techniques need a large amount of data to train the system.

Model-based techniques, on the other hand, will use models that capture the dynamics of the system by establishing mathematical relationship between variables. In this study, equations that can characterize the evolution of travel time over space are used. Many model-based studies use the Kalman Filtering Technique (KFT) for estimation. Advantages of this approach are it's limited data requirement and suitability for real time implementation. In this study, a model-based approach that uses data from just two previous buses will be compared with the ANN technique.
Thus, the objective of the present study is to predict the travel time of buses under Indian traffic conditions by using ANN and a model-based approach using KFT by providing the appropriate inputs to the models. The study compares the performance of a data driven approach with a model based method that requires only minimal data for bus travel time/arrival time prediction.

LITERATURE REVIEW

Since the present study focuses on the use of ANN and model based approaches for the bus arrival prediction, some of the previous works in this area are reviewed. Jeong et al. (2) reported a bus travel time prediction model based on ANN taking into account arrival time, dwell time and schedule adherence. Wang et al. (3) used support vector regression taking departure time from the stops as inputs to reflect the traffic conditions. The developed model also used historical bus travel time data, parameters of traffic conditions along bus route, and route specific parameters to predict future bus travel time. Liu et al. (4) developed a state space neural network model for bus arrival time prediction. Chien et al. (5) used two ANN models, link based and stop based, for bus arrival prediction. Park et al. (6) predicted link travel time by spectral basis artificial neural network (SNN) for one through five time period ahead for same vehicle. Dailey (7) used a combination of Automatic Vehicle Location (AVL) and historic database to predict travel time by using KFT and statistical analysis. Cathey (8) used bus travel time data as inputs to predict the same by using KFT that involved three components namely tracker, filter and predictor. Shalaby (9) used a combination of AVL and Automatic Passenger Counter (APC) data to predict travel time by using KFT. Nanthawichit et al. (10) used a combination of Global Positioning System (GPS) equipped probe vehicles and loop detectors data to estimate travel time by using KFT. The performance of the proposed methods were compared with historical data based, regression and ANN models separately.

All the studies discussed above dealt with homogeneous traffic conditions. Only a limited number of studies were reported under heterogeneous traffic conditions. Rama Krishna et al. (11) used 25 trips of bus travel time to develop ANN and Multiple Linear Regression (MLR) models. Vanajakshi et al. (12) used preceding two bus trips data collected by using GPS to predict next bus travel time by using KFT. Padmanabhan et al. (13) extended the above study by incorporating the delays in the model. Kumar (14) proposed a statistical methodology to find out patterns in the data and used them as input to predict the next bus travel time by using KFT. The present study will follow the above study to identify the travel time data most suited as inputs and use them to develop an ANN model to predict the bus arrival time. The performance of such a data driven technique will be compared with a model based approach with lower data requirement (using previous two buses data).

DATA COLLECTION, EXTRACTION AND ANALYSIS

For the purpose of collecting data, GPS equipped Metropolitan Transport Corporation (MTC) buses in the city of Chennai, India, were used. The test bed chosen for the present study is an MTC bus route, 5C, which connects the Parry’s bus depot in the northern part of the city to the Taramani bus depot in the southern part of the city. It has a route length of 15km with varying land use. Figure 1 illustrates the study route with bus stop details and distances between the bus stops are tabulated in Table 1.
Table 1 Distance between Bus Stops

<table>
<thead>
<tr>
<th>S.No</th>
<th>Bus stop name</th>
<th>Distance between bus stops (km)</th>
<th>Cumulative distance from the initial bus stop (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Parry’s Corner</td>
<td>0.00</td>
<td>0.00</td>
</tr>
<tr>
<td>2</td>
<td>Central Railway Station</td>
<td>0.93</td>
<td>0.93</td>
</tr>
<tr>
<td>3</td>
<td>P. Orr &amp; Sons</td>
<td>1.68</td>
<td>2.61</td>
</tr>
<tr>
<td>4</td>
<td>Wesley High School</td>
<td>2.70</td>
<td>5.31</td>
</tr>
<tr>
<td>5</td>
<td>C.I.T. Colony</td>
<td>2.34</td>
<td>7.65</td>
</tr>
<tr>
<td>6</td>
<td>Adyar Gate</td>
<td>2.04</td>
<td>9.69</td>
</tr>
<tr>
<td>7</td>
<td>Kotturpuram</td>
<td>1.96</td>
<td>11.65</td>
</tr>
<tr>
<td>8</td>
<td>Women’s Polytechnic College</td>
<td>2.21</td>
<td>13.86</td>
</tr>
<tr>
<td>9</td>
<td>Taramani</td>
<td>1.43</td>
<td>15.29</td>
</tr>
</tbody>
</table>

FIGURE 1 ROUTE MAP ALONG WITH BUS STOP DETAILS.

The collected GPS data include the ID of the GPS unit, time stamp, and latitude and longitude of the location at which the entry was made. Real time communication of this data was made possible through General Packet Radio Service (GPRS). The collected data is stored using Sequential Query Language (SQL) database encompassing all trips in a day. The average headway between two consecutive vehicles in this route is around 45 minutes. The data were collected for every 5 seconds from 6 AM to 8 PM for 45 days in the months of January and February 2013. The distance between two consecutive entries was calculated by using the Haversine formulae (15), which gives the great circle distances between two points on a sphere from their latitudes and longitudes as
\[ \text{Distance}(d) = 2r \arcsin\sqrt{\text{haversine}(\varphi_2 - \varphi_1 + \cos\varphi_1\cos\varphi_2 \text{haversin}(\lambda_2 - \lambda_1))}, \quad (1) \]

where \( r \) is the radius of the earth (6378.1 km), \( \varphi_1, \varphi_2 \) indicate the latitude of point 1 and point 2, \( \lambda_1, \lambda_2 \) indicate the longitude of point 1 and point 2. Thus, the processed data consists of the travel times and the corresponding distance between consecutive locations for all the buses. The entire section was divided into 150 subsections each of 100m length and the corresponding time taken to cover each subsection was calculated by using the linear interpolation technique. The outliers in the data were removed by keeping the lower bound permissible value as 5th percentile travel time while the higher bound permissible time was taken as 95th percentile travel time for each subsection.

**METHODOLOGY**

**Artificial Neural Network (ANN)**

A neural network is a massively parallel distributed processor made up of simple processing units that have a natural propensity for storing experimental knowledge and making it available for later use (16). This model basically tries to replicate how our brain works, its learning process and response to new sets of events. There are three layers in an ANN namely the input layer, the hidden layer and the output layer as shown in Figure 2. A basic unit of the connection is called a neuron, which is connected by other neurons through synaptic weights. Based on the desired target value, the network is trained, i.e., the weight matrix is updated after every iteration of the algorithm. Thus, as the number of iterations increases, the predicted output value matrix shifts closer to that of the target value.

![Figure 2: Basic Diagram of Neural Network of the Model Used in the Present Study](image)

Since the problem under study is a supervised learning problem, a multi-layer feed-forward network with Levenberg-Marquardt back propagation algorithm is used for training. This algorithm is considered to be one of the fastest method for training moderate-sized feed-forward neural networks which may range up to several hundred weights (16). It also has an
efficient implementation in MATLAB software. A hyperbolic tangent sigmoid function was used as the transfer function for both the hidden layer and the output layer.

The ANN model requires a good dataset of the desired output with corresponding inputs, making up the training set. After training, the model is simulated with a new input data set to check its efficacy. The predicted value obtained after simulation and the actual target value are compared and error percentage is found out. The total error is quantified using Mean Absolute Percentage Error (MAPE), which is

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^{N} \frac{x_p - x_a}{x_a},$$  

where $x_a$ is the actual value and $x_p$ is the predicted value. It is decided to use the six most correlated trips’ travel time on that particular stretch for each day as the inputs for the neural network. Thus, the numbers of nodes in the input and output layers are 6 and 1 respectively. The median analysis of trip acceptance ratio is used to select the number of inputs for the neural network model. The acceptance ratios of day wise pattern and trip-wise pattern are arranged in descending order in Table 2a. Now, the median of correlated value for all the seven days are analyzed which gives a value of 75% and above for top 6 correlated trips which is acceptable. One hidden layer with 7 neurons is identified as the best option by trial and error and checking error plots. The training rule for the ANN used in the study is

$$x_{k+1} = x_k - \left( J^T J + \mu I \right)^{-1} J^T e,$$  

where $J$ is the Jacobian matrix that contains first derivatives of the network errors with respect to the weights and biases, and $e$ is a vector of network errors. The Jacobian matrix can be computed through a standard back propagation technique (16). To test the performance of the above method, seven days’ data from 29th January 2013 to 4th February 2013 were taken. On the other hand, to identify the relevant inputs, a pattern analysis was carried out using two weeks data from 29th January 2013 to 12th February 2013. Since, the headway between two consecutive buses is approximately 45 minutes; one trip per every one hour was used from 6 AM to 8 PM. The Z-test for the mean of population of differences for paired samples was conducted for the hypothesis testing at 5% level of significance (14). For analyzing the daily pattern, each trip was compared with the previous seven days’ corresponding trips with the same starting time. For analyzing trip-wise pattern, each output trip was compared with the previous five trips that happened within the same day. A ratio has been calculated between the number of times the claimed null hypothesis is accepted to the total number of times the hypothesis tested for each case. If the acceptance ratio is high, we can conclude that input is significant in predicting the output trip (current trip). The pattern analysis results obtained for all days of the week are shown in Table 2A and the corresponding trips are shown in Table 2B.

**TABLE 2A: Acceptance ratios as per ranking**

<table>
<thead>
<tr>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
<th>Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.806</td>
<td>0.839</td>
<td>0.821</td>
<td>0.867</td>
<td>0.857</td>
<td>0.875</td>
<td>0.732</td>
<td>0.839</td>
</tr>
<tr>
<td>0.806</td>
<td>0.806</td>
<td>0.821</td>
<td>0.822</td>
<td>0.844</td>
<td>0.857</td>
<td>0.722</td>
<td>0.821</td>
</tr>
<tr>
<td>0.768</td>
<td>0.806</td>
<td>0.806</td>
<td>0.800</td>
<td>0.843</td>
<td>0.804</td>
<td>0.714</td>
<td>0.804</td>
</tr>
<tr>
<td>0.750</td>
<td>0.806</td>
<td>0.806</td>
<td>0.786</td>
<td>0.800</td>
<td>0.786</td>
<td>0.696</td>
<td>0.786</td>
</tr>
</tbody>
</table>
Table 2B: Pattern Analysis Results for ANN model

<table>
<thead>
<tr>
<th>Rank</th>
<th>Sunday</th>
<th>Monday</th>
<th>Tuesday</th>
<th>Wednesday</th>
<th>Thursday</th>
<th>Friday</th>
<th>Saturday</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>t-1</td>
<td>d-7</td>
<td>d-7</td>
<td>t-1</td>
<td>d-7</td>
<td>d-7</td>
<td>d-7</td>
</tr>
<tr>
<td>2</td>
<td>t-2</td>
<td>t-1</td>
<td>d-1</td>
<td>t-2</td>
<td>t-1</td>
<td>d-1</td>
<td>t-2</td>
</tr>
<tr>
<td>3</td>
<td>d-7</td>
<td>t-2</td>
<td>t-1</td>
<td>d-7</td>
<td>d-1</td>
<td>d-2</td>
<td>d-1</td>
</tr>
<tr>
<td>4</td>
<td>t-4</td>
<td>t-3</td>
<td>t-4</td>
<td>d-2</td>
<td>d-6</td>
<td>d-4</td>
<td>d-2</td>
</tr>
<tr>
<td>5</td>
<td>t-3</td>
<td>d-2</td>
<td>d-5</td>
<td>d-1</td>
<td>t-2</td>
<td>d-3</td>
<td>d-4</td>
</tr>
<tr>
<td>6</td>
<td>d-1</td>
<td>d-5</td>
<td>d-4</td>
<td>t-3</td>
<td>d-3</td>
<td>t-1</td>
<td>t-3</td>
</tr>
<tr>
<td>7</td>
<td>t-5</td>
<td>d-4</td>
<td>t-5</td>
<td>d-6</td>
<td>t-2</td>
<td>t-2</td>
<td>t-3</td>
</tr>
<tr>
<td>8</td>
<td>d-3</td>
<td>t-4</td>
<td>d-6</td>
<td>t-4</td>
<td>d-3</td>
<td>d-6</td>
<td>t-4</td>
</tr>
<tr>
<td>9</td>
<td>d-5</td>
<td>t-5</td>
<td>t-3</td>
<td>d-4</td>
<td>t-5</td>
<td>t-3</td>
<td>d-3</td>
</tr>
<tr>
<td>10</td>
<td>d-2</td>
<td>d-3</td>
<td>t-2</td>
<td>d-5</td>
<td>d-5</td>
<td>t-4</td>
<td>t-5</td>
</tr>
<tr>
<td>11</td>
<td>d-4</td>
<td>d-6</td>
<td>d-3</td>
<td>t-5</td>
<td>t-4</td>
<td>t-5</td>
<td>d-6</td>
</tr>
<tr>
<td>12</td>
<td>d-6</td>
<td>d-1</td>
<td>d-2</td>
<td>d-3</td>
<td>d-4</td>
<td>d-5</td>
<td>d-5</td>
</tr>
</tbody>
</table>

* (d-n) represents previous n° day same time trip and (t-n) represents previous n° trip within the same day.

From this, the six most relevant trips were used as inputs for each day (75% or more acceptance ratio). Data were normalized before using based on the formula

$$\frac{x_k(i) - \min_k(i)}{\max_k(i) - \min_k(i)}$$

where $$x_k(i)$$ is the $$i$$th element of column ‘$$x$$’ in the dataset, and $$\max_k(i)$$ and $$\min_k(i)$$ are respectively minimum and maximum value of that particular data considering training and testing datasets. The training stops when any of the following conditions are met:

1. The maximum number of epochs (repetitions) is reached, which was chosen as 600.
2. The maximum training time has exceeded.
3. Performance has been minimized to the goal which is taken as 0.001.
4. The performance gradient falls below minimum gradient which is 1e-10.
5. The value of ‘$$\mu$$’, which signifies the rate of convergence of adaptive algorithm, exceeds its maximum which is 1e10.

One of the problems in ANN training is over fitting. Over fitting reduces the generalizing capability of the network resulting in inflation in the output value with unseen data. To curb the problem of over fitting, regularization of the performance function was carried out. This was done by adding a term that consists of the mean of the sum of squares of the network weights...
and biases (msw) to the typical performance function, Mean Square Error (MSE). Thus, the new performance function become

\[
\text{New performance function} = \alpha \text{mse} + (1 - \alpha) \text{msw},
\]

(5)

In this study, the value of \( \alpha \) is taken as 0.5. Since the weight matrix is randomly initialized for every run, a total of 7 runs were carried out for each training and from these seven runs, an ensemble averaging method which is a type of committee machine, is used to find out the final output. In this method, the output produced by 7 different trained networks are combined and linearly averaged. The motivation of using this technique is due to the fact that differently trained networks (caused by random weights) converge to different local minima on the error surface, and the overall performance is improved by combining the outputs (16).

**Model Based Approach**

A model based approach using KFT was used in this study for comparing the performance with ANN. The KFT (17) can be used to estimate state variables, which are used to characterize systems/processes that are described by state space models. The implementation of the Kalman filter requires information regarding the system’s dynamics, statistical information of the system disturbances and measurement errors. It uses the model and the system inputs to predict the \textit{a priori} state estimate and uses the output measurements to obtain the \textit{a posteriori} state estimate. Overall, it is a recursive algorithm, so that new measurements can be processed when they are obtained. It needs only the current instant state estimate, current input and output measurements to calculate next instant’s state estimate. The selected approach (12) has minimal data requirement and the aim of the study is to find out whether the use of a large data base and a corresponding data driven approach improve the prediction accuracy significantly. The evolution of travel time between the various subsections was assumed to be

\[
x(k + 1) = Ax(k) + w(k),
\]

(6)

where, \( A \) is a parameter which relates the travel time in the \( k^{th} \) subsection to the travel time in the \( (k+1)^{th} \) subsection, \( x(k) \) is the travel time taken for covering the given subsection \( k \) and \( w(k) \) is the associated process disturbance with the \( k^{th} \) subsection. The measurement process was assumed to be governed by

\[
z(k) = x(k) + v(k),
\]

(7)

where \( z(k) \) is the measured travel time in a given subsection \( k \) and \( v(k) \) is the measurement noise. It was further assumed that \( w(k) \) and \( v(k) \) are zero mean white Gaussian noise signals with \( Q(k) \) and \( R(k) \) being their corresponding variances. Thus, two sets of data are required to implement the above scheme - one set of data for the time update equations to calculate the parameter ‘\( A \)’ and another data set to be used in the measurement update equations to generate the \textit{a posteriori} travel time estimate. So, the data obtained from first bus (PV1) were used to obtain the value of \( A \) for each subsection and the data from second bus (PV2) were used to obtain the \textit{a posteriori} travel time estimate of next vehicle (TV).

The steps in the algorithm were as follows:
1. The entire section of travel between origin and destination was divided into N subsections of equal length (100 m).

2. The travel time data from PV1 was used to obtain the value of $A$ by making the assumption that there exists a relation between neighboring subsections and the relation was given as

$$A = \frac{x_{PV1}(k+1)}{x_{PV1}(k)}, k = 1, 2, 3, \ldots \ldots (n-1) \quad (8)$$

3. Let $x_{TV}$ be the travel time taken by the test vehicle (the vehicle for which the travel time needs to be predicted) to cover the given subsection. For the first section, one cannot predict the travel time of test vehicle (TV). So, it was assumed that

$$E[x_{TV}(1)] = \hat{x}(1), \quad (9)$$

$$E[x_{TV}(1) - \hat{x}(1)^2] = P(1), \quad (10)$$

where $\hat{x}(k)$ is the estimate of travel time of the TV in the $k^{th}$ subsection.

4. For $k = 2, 3, \ldots, (N-1)$, the following steps were performed:

a. The priori estimate of the travel time was calculated by using $\hat{x}^{-}(k+1) = A\hat{x}^{-}(k)$, where the superscript $'-'$ denotes the a priori estimate and the superscript $'+'$ denotes the a posteriori estimate.

b. The a priori error variance (denoted by $P^{-}$) was calculated using

$$P^{-}(k+1) = AP^{-}(k)A + Q(k), \quad (11)$$

c. The Kalman gain (denoted by $K$) was calculated by using

$$K(k+1) = P^{-}(k+1)[P^{-}(k+1) + R(k+1)]^{-1}, \quad (12)$$

d. The a posteriori travel time estimate and error variance were calculated using, respectively,

$$\hat{x}^{+}(k+1) = \hat{x}^{-}(k+1) + K(k+1)[z(k+1) - \hat{x}^{-}(k+1)], \quad (13)$$

$$P^{+}(k+1) = [I - K(k+1)]P^{-}, \quad (14)$$

Thus, the objective here is to predict the travel time of the TV using the travel time obtained from previous two bus data (PV1 and PV2) when the TV is in the $k^{th}$ subsection.

RESULTS AND DISCUSSIONS

A comparison of performance of the proposed methods was carried out section wise as well as trip wise. The prediction accuracy was calculated using MAPE. The performance of ANN and KFT were compared using the testing data set from 29th January to 4th February, 2013. The bus travel time for each 100 m subsection of the entire section was predicted using this scheme. It has to be noted that for this scheme, the data from the previous 2 buses alone are required as inputs.

Figure 3 shows a sample result where the predicted travel time from ANN is shown against the actual travel time for one particular section. It can be observed that the prediction
values are following the trends in the measured data. The MAPE for this case was observed to be 23.23%. Figure 4 shows the corresponding MAPE for all the subsections. A similar analysis was carried out using the model based approach and the results obtained are also shown in Figure 4. The error varies from 13.93% to 44.52% for ANN and 16% to 42% for the model based approach. It can be seen that the results are comparable for both the data driven and model based approaches, with a slight advantage to ANN. However, considering the data base requirement for ANN and the minimal data requirement for model based approach, the performance of both can be considered comparable. Especially with agencies that have a minimal data base, the model based approach can be a viable option.

![Figure 3](image3.png)

**FIGURE 3** MEASURED TRAVEL TIME AND PREDICTED TRAVEL TIME USING ANN.

![Figure 4](image4.png)

**FIGURE 4** COMPARISON BETWEEN MACHINE LEARNING AND MODEL BASED APPROACHES.

The performance was also checked for specific trips across the stretch. Sample results on 31st January 2013 for one peak and one off-peak trips’ using ANN are shown in Figures 5A and 5B. It can be observed that the model performs well for all the subsections for both peak and off-peak conditions. Corresponding plots using model based approach are shown in Figure 6A and
6B. Errors were quantified using MAPE and are shown in Table 3 for various trips on a selected day. It can be observed that ANN performed better than KFT in most of the cases.

**FIGURE 5A PEAK TIME PERFORMANCE – TRIPWISE USING ANN.**

**FIGURE 5B OFF PEAK TIME PERFORMANCE – TRIPWISE USING ANN.**

**FIGURE 6A PEAK TIME PERFORMANCE – TRIPWISE USING MODEL BASED APPROACH.**
SUMMARY AND CONCLUSIONS
The main aim of Advanced Public Transportation System (APTS) is to attract passengers towards public transport, and thus help to reduce the congestion on urban roads. However, for this to happen in practice, the bus service should be made more attractive and one option for that is to provide reliable information about bus arrival to passengers. However, the reliability of the information provided to passengers mainly depends on the prediction technique used and the input data used in the same. The present study compared the performance of two commonly used prediction techniques, one data driven and the second with minimal data requirement for bus arrival prediction. The data driven technique selected is the Artificial Neural Network method and a model based approach using Kalman Filtering Technique with minimal data requirement of just two previous buses is used for less data demanding technique. To make sure that the data used is the most significant one, pattern analysis using statistical methods was carried out. The performances of both methods are evaluated using data collected for a period of one week from the field. The comparison of both methods showed a better performance from ANN compared to the model based KFT. However, the caveat of using ANN over KFT is the requirement of a large dataset for network training. On the other hand, the model based KFT, which gave comparable
performance, require very minimal data and hence will be suitable in cases where a good data base is not available.

ACKNOWLEDGEMENT
The authors acknowledge the support for this study as a part of the sub-project CIE/10-11/168/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.

REFERENCES

