

# Artificial Neural Network Travel Time Prediction Model for Buses Using Only GPS Data

Zegeye Kebede Gurmu, Teague Nall & Perkins, Inc.  
Wei (David) Fan, University of North Carolina at Charlotte

## Abstract

*Real-time and accurate travel time information of transit vehicles is valuable as it allows passengers to plan their trips to minimize waiting times. The objective of this research was to develop a dynamic artificial neural network (ANN) model that can provide accurate prediction of bus travel times to give real-time information at a given downstream bus stop using only global positioning system (GPS) data. The ANN model is trained off-line but can be used to provide real-time travel time information. To achieve this, care was taken in selecting a unique set of input-output combinations for prediction. The results obtained from the case study are promising to implement an Advanced Public Transportation System (APTS). The performance of the proposed ANN model was compared with a historical average model under two criteria: prediction accuracy and robustness. It was shown that the ANN outperformed the average approach in both aspects.*

## Introduction

Growing traffic congestion has posed threat to the quality of life of people in many countries over the past few decades. Congestion in general leads to a decrease in accessibility and mobility, travel time loss, and air pollution. Many different solution techniques have been suggested, including demand-side (congestion pricing, traffic management, etc.) and supply-side (constructing more roads, adding lanes,

etc.) or their integration, for mitigation of congestion. One potential supply-side tactic is to improve and expand public transportation service (Houghton et al. 2009; Dewan and Ahmad 2007). Public transport service can be enhanced by providing travelers with reliable travel information through the help of an Advanced Public Transportation System (APTS), which is one component of Intelligent Transportation Systems (ITS) (Houghton et al. 2009; Vanajakshi et al. 2009). Travel time information is the most preferred information by travelers. The provision of timely and accurate transit travel time is vital because it attracts more people to public transport and increases the contentment of transit users (Jeong and Rilett 2004). However, real-time travel time information cannot be measured directly. Therefore, to provide passengers with this information, mathematical models that can predict travel time with reasonable accuracy are required. A variety of prediction models for forecasting traffic states such as travel time and traffic flow have been developed over the years. The most widely-used bus travel time prediction models can be classified into four categories, which are discussed below.

- 1. Historical Average Models** (Jeong and Rilett 2004; Farhan et al. 2002; Ramakrishna et al. 2006). Historic data-based average prediction models give the current and future travel time from the historical bus travel time of previous journeys, and the current traffic condition is assumed to remain stationary. Therefore, a model of this kind is reliable only when the traffic pattern in the area of interest is relatively stable or where congestion is minimal, e.g., rural areas.
- 2. Regression Models** (Jeong and Rilett 2004; Ramakrishna et al. 2006; Patnaik et al. 2004; Chien et al. 2002; Chen et al. 2004; Shalaby and Farhan 2003). Regression models predict and explain a dependent variable with a linear function formed by a set of independent variables. Unlike historical data-based prediction models, these are able to work satisfactorily under unstable traffic condition. Regression models usually measure the simultaneous effects of various factors, which are independent between one and another, affecting the dependent variable. Patnaik et al. (2004) proposed a set of multiple linear regression models to estimate bus arrival times using data collected by automatic passenger counters (APC). Distance, number of stops, dwell times, number of boarding and alighting passengers, and weather descriptors were used as independent variables. The study indicated that the models could be used to estimate bus arrival time at downstream stops. Jeong and Rilett (2004) and Ramakrishna et al. (2006) also developed multiple linear regression

models using different sets of inputs. Both studies indicated that regression models are outperformed by other type of models. However, one great advantage of multiple linear regression models is that they can reveal which inputs are less or more important for predicting travel times. For example, Patnaik et al. (2004) discovered that weather was not an important input in their model. Ramakrishna et al. (2006) also found out that bus stop dwell times from the origin of the route to the current bus stop in minutes and intersection delays from the origin of the route to the current bus stop in minutes are less important inputs. In general, the applicability of regression models is limited because variables in transportation systems are highly intercorrelated (Chien et al. 2002).

**3. Kalman Filtering Models** (Chien et al. 2002; Shalaby and Farhan 2003). Kalman filtering models have elegant mathematical representations (e.g., linear state-space equation) and the potential to adequately accommodate traffic fluctuations with time-dependent parameters (e.g., Kalman gain) (Chien et al. 2002). These models have been used extensively for predicting bus arrival time (Chien et al. 2002; Chen et al. 2004; Shalaby and Farhan 2003). Their basic function is to provide estimates of the current state of the system, but they also serve as the basis for predicting future values or for improving estimates of variables at earlier times, i.e., they have the capacity to filter noise (Kalman 1960).

**4. Machine Learning Models** (Jeong and Rilett 2004; Chen et al. 2004). Machine learning methods such as Artificial Neural Networks (ANN) can deal with complex relationships between predictors that can arise within large amounts of data, process non-linear relationships between predictors, and process complex and noise data. These models can be used for prediction of travel time without explicitly addressing the (physical) traffic processes. The ANN method is classified under this category. ANNs recently have been gaining popularity in predicting bus arrival time because of their ability to solve complex non-linear relationships (Jeong and Rilett 2004; Ramakrishna et al. 2006; Chien et al. 2002; Chen et al. 2004). ANNs, inspired by emulating the intelligent data processing ability of human brains, are constructed with multiple layers of processing units, named artificial neurons. The neurons contain activation functions (linear or nonlinear) and are highly interconnected with one another by synaptic weights. Information can be processed in a forward or feedback direction through fully or partially

connected topologies. Meanwhile, the synaptic weights can be adjusted to map the input-output relationship for the analyzed system automatically through a learning process (Hagan et al. 1996). However, results obtained using these models for one location are (typically) not transferrable to the next, due to location-specific circumstances (geometry, traffic control, etc.).

ANNs developed by different researchers in predicting bus travel time differ in their input-output combinations. In addition, they use explanatory variables such as flow, speed, weather, distance, etc., as inputs. However, few research efforts have been made in the area of bus travel time prediction using GPS technology in the absence of data on the stated traffic-stream variables. The purpose of this paper is, therefore, to explicitly consider arrival and departure time information at stops collected via GPS technology to predict bus travel time. The proposed bus travel time prediction model is based on historical arrival/departure time patterns and real-time arrival/departure time information. Hence, nonlinear correlations between travel times can be captured to predict bus travel time at downstream bus stops. The proposed ANN model is trained off-line but can be used to provide real-time travel time information. To achieve this objective, care was taken in selecting a unique set of input-output combinations for prediction while maintaining the reproducibility of the model.

## **ANN Model Structure**

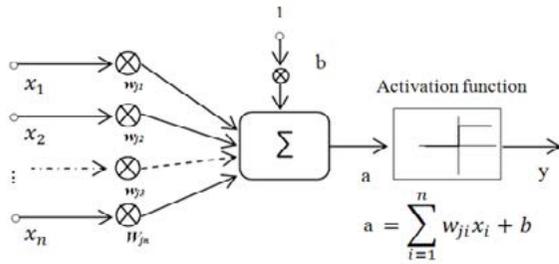
### ***ANN***

ANNs learn from examples and capture subtle functional relationships among data even if the underlying relationships are unknown or hard to explain. Thus, ANNs are well-suited for problems whose solutions require knowledge that is difficult to specify. Another advantage of ANNs is that they can generalize. After learning the data fed into them (a sample or example), ANNs can often correctly infer the unseen part of a population, even if the example data contain noisy information. However, to gain the maximum benefit from a neural network, there should be enough data or observations (Zhang et al. 1998).

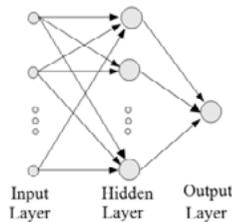
### ***Network Architecture***

Many different ANNs have been proposed in the past few decades for forecasting purposes. The most popular connected multilayer perceptron (MLP) neural network architecture was chosen in this study because it can approximate almost any function if there are enough neurons in the hidden layers, i.e., it has a very good capability of arbitrary input-output matching (Haykin 1999). It is also easy to

implement. The ANN architecture is typically composed of a set of nodes and connections arranged in layers. In this study, three layers were used: input, hidden, and output. The first layer is an input layer when external information is received. The last layer is an output layer where the problem solution is obtained. Usually, one or two hidden layers are used in between the first and last layers to predict reasonably well. The actual processing in the network occurs in the nodes of the hidden layer and the output layer. The input layer is where the data vector is fed into the network. It then feeds into the hidden layer, which, in turn, feeds into the output layer. The connections are typically formed by connecting each of the nodes in a given layer to all of the neurons in the next layers. The hidden layer generates the weight of these connections and the bias parameter during the training process. It is the hidden nodes in the hidden layer that allow the neural network to detect the feature to capture the pattern in the data, to perform nonlinear mapping between input and output variables. A single hidden layer has proved to be sufficient for ANNs to approximate any nonlinear functions (Zhang et al. 1998). A suitable number of nodes in the hidden layer need to be determined by experiment. A fully-connected MLP with one hidden layer is presented in Figure 1.



a) Neuron



b) FeedForward Network

**Figure 1. Multilayer perceptron (MLP) neural network architecture**

**ANN Model Development**

Although the basic training procedure of ANNs is the same, the accuracy of the result is greatly dependent upon the type of input/output combinations. The input variables are presented in such a way that the function signal appearing at the output of neuron j is computed as:

$$Y_j = \psi(X_1, X_2, X_3, \dots, X_n) \tag{1}$$

This can be written formally as:

$$Y_j = \psi_j \left( \sum_{i=1}^m w_{ji} X_i + b_j \right) \tag{2}$$

where,

$m$  is number of inputs applied to neuron j

$\{X_i\}$  is set of input variables of neuron j

$Y_j$  is output of the j<sup>th</sup> neuron

$w_{ji}$  is the synaptic weight connecting the i<sup>th</sup> input to the j<sup>th</sup> neuron,

$b_j$  is error term and

$\psi_j (\cdot)$  is an activation function

The activation function  $\psi_j (\cdot)$  is usually needed to introduce nonlinearity into the network. It determines a nonlinear relationship between inputs and outputs of a node and a network. The sigmoidal functions such as logistic and hyperbolic tangent functions (tanh) are the most common choices. Functions such as tanh or arctan that produce both positive and negative values tend to yield faster training than functions that produce only positive values such as logistic in practice (Haykin 1999; Karlik and Olgac 2010). Hence, in this study, the tanh function is used to scale inputs and targets to (-1, 1). This function is given by:

$$\psi(X) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{3}$$

The training procedure chosen was the most commonly used back-propagation algorithm, which is arguably the most popular algorithm for transportation use (Jeong and Rilett 2004; Chien et al. 2002; Chen et al. 2004). The objective of the training process is to improve weights  $w_{ji}$  that minimize the mean squared error

(MSE) (Haykin 1999). The average error  $\varepsilon_{avg}$  at the output of neuron  $j$  at iteration  $n$  for  $N$  number of examples in the training set is defined by:

$$\varepsilon_{avg} = \frac{1}{2N} \sum_{n=1}^N \sum_j (d_j(n) - y_j(n))^2 \quad (4)$$

where  $d_j(n)$  is the desired output. The back-propagation algorithm applies a correction  $w_{ji}$  to the synaptic weight  $\Delta w_{ji}$ , which is proportional to the partial derivative  $\partial \varepsilon_{avg} / \partial w_{ji}$ . That is:

$$\Delta w_{ji} = -\eta \frac{\partial \varepsilon_{avg}}{\partial w_{ji}} \quad (5)$$

where  $\eta$  is the learning-rate parameter of the back-propagation algorithm. It has typically a value between 0.001 and 1.0. If smaller learning rate is considered, smaller changes to synaptic weight will occur. If, on the other hand, one makes a large learning rate, the resulting large changes in the synaptic weights will follow such a form that the network may become unstable (Haykin 1999). Therefore, a simpler method of increasing the rate of learning was proposed by including a momentum term in the above equation (Rumelhart 1986):

$$\Delta w_{ji}(n) = \alpha \Delta w_{ji}(n-1) + \eta \delta_j(n) X_i(n) \quad (6)$$

where  $\delta_j(n)$  is gradient and defined by:

$$\delta_j(n) = - \frac{\partial \varepsilon_{avg}}{\partial (\sum_{i=1}^m w_{ji}(n) X_i(n) + b_j)} \quad (7)$$

### **The ANN Prediction Algorithm**

Consider the bus route shown in Figure 2. Suppose a journey for a bus  $k$ , equipped with GPS, is initiated from stop 0 at a certain time of day interval  $t$  and the bus is currently at location  $c$ , which may or may not be a bus stop after passing stop  $i$ . It

is required to provide travel time information for a person at stop  $j$ . Therefore, the travel time information from the current bus location  $c$  to stop  $j$  can be calculated as:

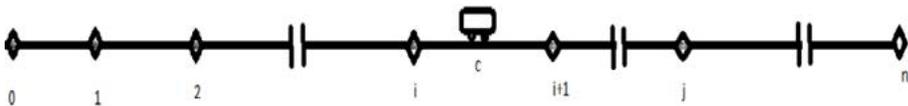
$$TT^{k}_{cj} = TT^{k}_{ij} - TT^{k}_{ic} \tag{8}$$

Where,

$TT^{k}_{cj}$  is predicted bus travel time from location  $c$  to stop  $j$  for bus  $k$

$TT^{k}_{ij}$  is predicted bus travel time between stop ‘ $i$ ’ and ‘ $j$ ’ for bus  $k$

$TT^{k}_{ic}$  is travel time to point  $c$  after passing stop ‘ $i$ ’ for bus  $k$



**Figure 2. Hypothetical bus route**

$TT^{k}_{ic}$  value can be determined by deducting departure time at stop  $i$  from the current time when travel time information is requested.  $TT^{k}_{ij}$  is determined using ANN. As has been indicated earlier, a unique set of inputs and output that can be obtained from GPS are considered. The input variables chosen are:

- X1 = the time of day interval  $t$
- X2= coded id number of bus station  $i$
- X3= coded id number of bus station  $j$
- X4= the travel time taken from stop 0 to  $i$

The output  $Y$  will be predicted bus travel time to reach stop  $j$  from stop  $i$ , which is equivalent to  $TT^{k}_{ij}$  in equation (8).

Training and learning functions are mathematical procedures used to automatically adjust the network’s weights and biases (Mathworks). The training function dictates a global algorithm that affects all the weights and biases of a given network. The learning function can be applied to individual weights and biases within a network. Neural networks are trained so that a particular input leads to a specific target output. The network is adjusted based on a comparison of the output and the target until the network output matches the target. In this study, MATLAB

was used to train a network using the aforementioned set of inputs and a target variable, which is the observed travel time between stops  $i$  and  $j$ . Let the trained network be NEURAL\_NET. The function *sim* simulates a network (Mathworks). It takes the network input vector  $\{X_n\}$ , and the network object NEURAL\_NET, and returns the network output  $Y$ . That is:

$$TT_{ij}^k = Y = \text{sim}(\text{NEURAL\_NET}, \{X_n\}) \quad (9)$$

Usually, the input/target data sets need to be normalized before training. This is because the contribution of an input will depend heavily on its variability relative to other inputs. If one input has a range of 0 to 1, while another input has a range of 0 to 2,000,000, then the contribution of the first input will be swamped by the second input. Therefore, it is essential to rescale the inputs so that their variability reflects their importance, or at least is not in inverse relation to their importance. The choice of range to which inputs and targets are normalized depends mainly on the activation function  $\psi_j(\cdot)$  of output nodes, with typically (0, 1) for logistic function and (-1, 1) for hyperbolic tangent function.

It is worth mentioning that the choice of input variables makes the algorithm possible to predict dynamically even when the neural network is trained off-line. The first input variable, for example, represents time of the day, which accounts for the variability of travel time between different hours of the day. The last input variable also takes the current travel time information from the origin up to the recently-visited stop.

In the development of a program for a neural network, several steps of algorithms presented above are followed. However, care should be taken while preparing input/target variables for training since one has to deal with a large amount of data sets. Another problem is when to stop the training. On one hand, under-training may occur and the training patterns may not be sufficiently representative of the true population. Hence, the network needs to be exposed to enough examples. On the other hand, over-training could cause memorization, where the network might simply memorize the data patterns and might fail to recognize other set of patterns. Thus, early stopping at the proper time is recommended to ensure that the network learns accordingly.

## Case Study

GPS data were available from November 2008 to May 2009 for buses in Macae, Brazil. Bus line LT11 was chosen for the case study because it had the largest num-

ber of data sets compared to the other bus lines (1M+ records). The route, shown on Figure 3, has 35 stops, numbered 0 to 34. Data were collected using Automatic Vehicle Location (AVL) systems. In these systems, GPS receivers are usually interfaced with Global System for Mobile Communications (GSM) modems and placed in the buses. The systems basically record point locations in latitude-longitude pairs, speeds of the buses, date, and time. Arrival and departure time records at each bus stop are the most important.

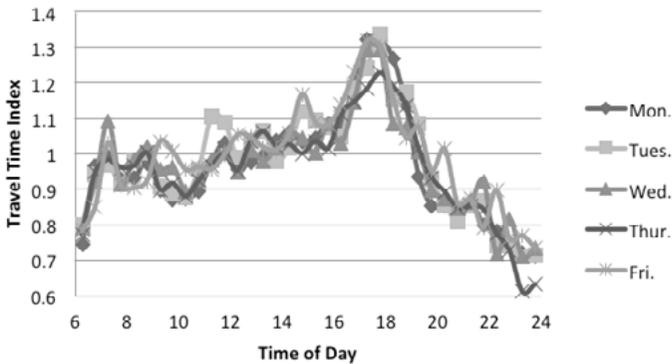


Figure 3. Bus route LT11, Macaé, Brazil

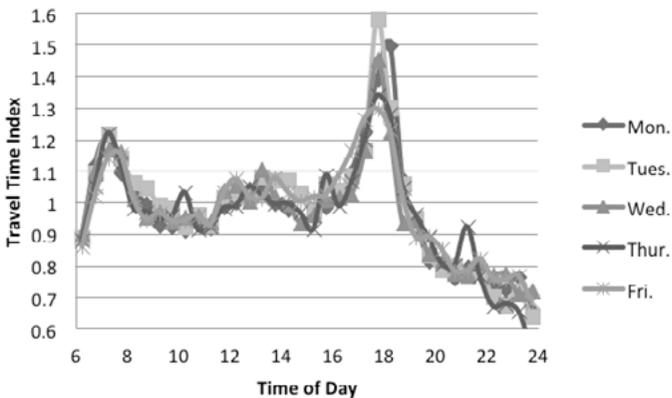
### Preliminary Analysis

To calculate travel times, the travel times were pooled together in 30-minute intervals (e.g., 06:00–06:30, 06:30–07:00, etc.) and the average value of each interval was determined. This means that all data sets needed to be clustered by time period, because transit vehicles have different departure times by time period and, also, there might not be bus schedules, resulting in different travel patterns. Usually, the average travel time from historical data can be taken as a baseline prediction model, which can be used for comparison purposes later.

Figure 4 shows the travel time index as function of time of the day, i.e., the ratio of the average travel time per weekday and the average travel time over all days. The different lines correspond with different workdays, i.e., Monday–Friday. The upper panel demonstrates the line LT11 in the northbound direction (from stop 0 to stop 34), and the lower panel shows the southbound direction. From Figure 4, it can be seen that there is a significant variation in (average) travel time over different times of the day. In the evening rush hour (17:00–18:00), the travel times are about 30 percent higher than the average (over all time periods and all days). During the morning rush hour (around 07:00–07:30), travel times are about 20 percent higher than average for the southbound direction. Travel time variations during the day, thus, are significant and should be taken into account. This explains why we chose time of day as input variable in our proposed model.



a) LT11 northbound direction

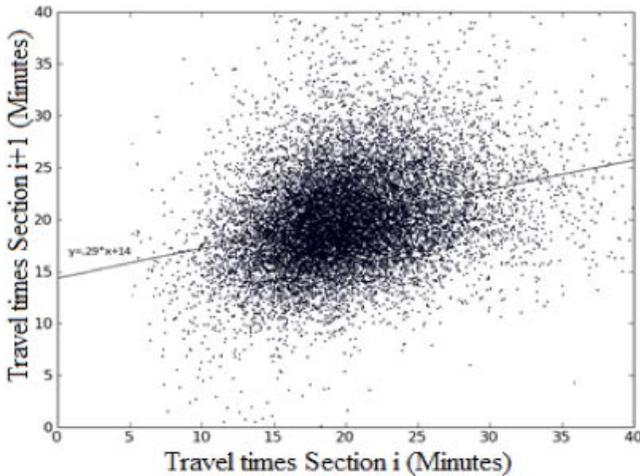


b) LT11 southbound direction

**Figure 4. Travel time variation over time of day**

Another observation is that travel time distributions over different days of the week are nearly the same without significant differences, as indicated and confirmed by performing Duncan’s multiple range test. Thus, these variations were not considered in the model development.

It is possible to improve prediction models from average values by considering errors or variations and correlations between different values of the variable under consideration (Thomas et al. 2010). For example, if a bus is slower during the first section of the trip, it is likely that it will also be slower on the second section of the trip. Therefore, we say that travel time of successive buses or of successive trip sections may be correlated. If these correlations are found, the information about the travel time of a previous bus or trip section can be used to update travel time prediction of the next bus or trip section. To illustrate this, correlation analysis was conducted between observed travel times of successive sections. First, the whole trajectory was divided into four approximately equal sections. The average travel time in each section was around 20 minutes. Figure 5 presents correlation results of travel times between successive sections.



**Figure 5. Correlation between travel times of successive sections**

As can be seen from Figure 5, it was not possible to get satisfactory linear relationships ( $R^2=0.09$ ) to improve the baseline prediction based on correlation. Hence, the proposed artificial neural network models were used to capture these highly nonlinear relationships and predict bus travel time at the downstream bus stops. Data reduction was done using Python(x, y) software, and the input/target data sets were prepared to be trained in MATLAB.

### ***Training, Test, and Validation Data Sets***

Training and test samples typically are required for building an ANN forecaster. The training sample is used for ANN model development and the test sample was adopted for evaluating the forecasting ability of the model or to measure its performance. A validation sample was also used to avoid the over-training problem or to determine the stopping point of the training process. The symptom of over-training is that the network performs well with data in the training set, while its performance over the test data set (those “unseen” by the network) starts to deteriorate. It is common to use one test set for both validation and testing purposes.

The first issue to deal with during ANN development is the division of the data into the training and test sets. Although there is no general procedure to do this, several factors, such as the data type and the size of available data, should be considered in the division. The whole data set was first sorted by week number. Then, the first 70 percent of the data set was taken as a training set and the next 30 percent as a testing set. This division has been used by most researchers (Yu et al. 2006; Patricia and Robert 2005). Of the testing set, 20 percent of the data set was taken as a validation test. It is worth mentioning that different percentage combinations of training, test, and validation sets had been investigated, and it was possible to get a minimum mean square error using the above combinations.

### ***Training and Learning Functions***

The training function dictates a global algorithm that affects all the weights and biases of a given network. The learning function can be applied to individual weights and biases within a network. MATLAB offers a number of training and learning functions. In this study, two training functions and one learning function were used. Of 12 training functions, Jeong and Rilett (2004) found that the Bayesian Regularization training function and the Levenberg-Marquardt Back propagation training function outperformed the other 10 training functions. In this study, it was found that the Levenberg-Marquardt Back propagation training function, which is commonly used by most researchers (Jeong and Rilett 2004; Chien et al. 2002, Chen et al. 2004), outperformed the Bayesian Regularization training function. Jeong and Rilett (2004) also showed that there were no significant differences in the results from the 14 tested learning functions. Gradient descent with momentum weight and bias learning function were used in this study to make it consistent with proposed algorithm.

## Numerical Results

### Model Performance

After the prediction model was developed, it was necessary to evaluate its performance in terms of prediction accuracy. Since regression prediction models are not good option in the absence of traffic data, and since there are several variations of Kalman filtering models that may complicate the evaluation process, as mentioned previously, the proposed ANN model was compared only with that of the historical average travel time model. The Mean Absolute Percentage Error (MAPE) was used as the measure of model performance, representing the average percentage difference between the observed value (in this case, observed arrival times at a bus stop) and the predicted value (in this case, predicted arrival times at a bus stop).

$$MAPE = \frac{100}{n} \sum_{t=1}^n \left| \frac{Y_t^P - Y_t^{Obs}}{Y_t^{Obs}} \right| \tag{10}$$

where,

$Y_t^P$  is the predicted bus travel time from recent bus stop to target bus stop

$Y_t^{Obs}$  is the observed bus travel time

$n$  is the number of test sets

Figure 6 presents the average MAPE values (computed using Eq. [10]) for a range of observed travel times between stops. As can be seen, prediction of bus travel time in the study area could be given with an overall average MAPE value of 18.3 percent using the ANN model.

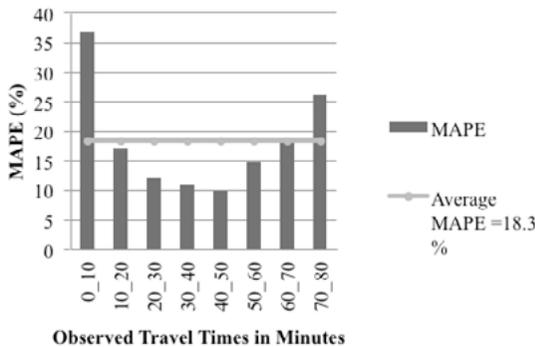


Figure 6. MAPE for a range of observed travel times

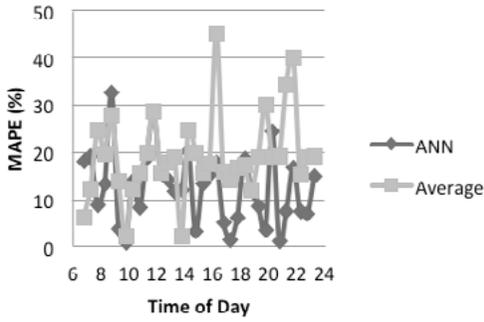
Around 25 percent of the test data set resulted in MAPE higher than the total average MAPE (i.e., 18.3%). Figure 6, thus, shows that only a few test data sets could be observed with a higher MAPE when the number of stops between the current bus location and the stop under consideration becomes too large or too small. It was discovered that the ANN model gave a better prediction of travel time when the station where the travel time information required is located at least five stops away from the current bus location. On one hand, for observed travel times with values of 20–50 minutes, prediction can be given with less than 10 percent error. In other words, it is possible to provide real-time travel time information for a person who is 20–50 minutes away from the current bus location with a maximum 10 percent error. On the other hand, when the observed travel time becomes larger, i.e., greater than 50 minutes, the error in prediction starts to increase. This could be due to the fact that when a bus travels longer in a trip, there is a high probability that it stops at many bus stops, especially when demand is high during peak hours. Therefore, a person who is 20–50 minutes away from the current bus location receives the optimal travel time information. The error in prediction for smaller travel times, for example, 5–10 minutes, was found to be higher even for the ANN. This indicates that for short distance trips there is higher travel time variability, as one might expect, due to a number of factors. For example, a bus may wait for one minute at a traffic light on a link where the bus usually takes two minutes to cross it. So, the error for this situation could be exaggerated and may reach up to 40 percent, which, in turn, will increase the overall MAPE. However, it should be acknowledged that this kind of travel time information may be less important as it is within an acceptable range of waiting time.

### **Model Comparison**

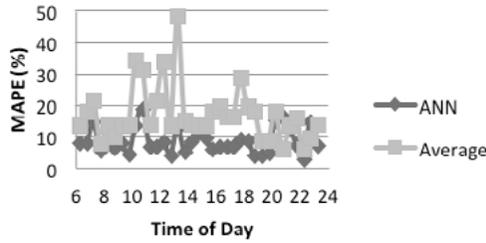
Three different sections were considered to show the comparison between the two models. The sections are defined for one direction, i.e., northbound, as shown in Table 1.

**Table 1. Sections Compared**

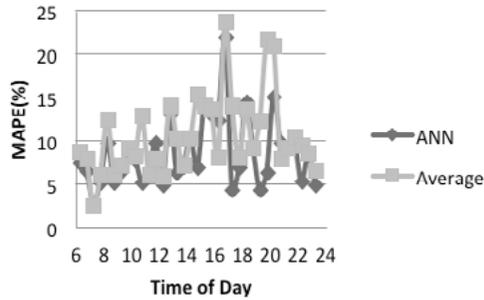
<b>Trajectory</b>	<b>Stop ID's</b>	<b>Remark</b>
Section 1	Stop 8 to Stop 16	Short section
Section 2	Stop 8 to Stop 26	Medium section
Section 3	Stop 0 to Stop 34	Long (whole section)



a) Short section (Section 1)



b) Medium section (Section 2)



c) Long section (Section 3)

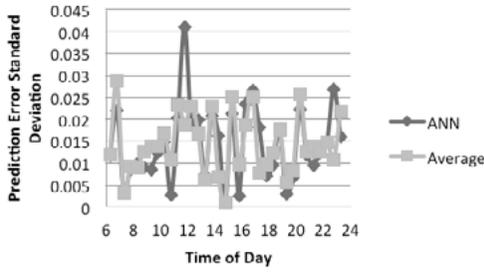
**Figure 7. MAPE vs. historical average**

Even though, there is no much difference in MAPE between the two models for large and small sections, the ANN outperformed the historical average model approach on more than 70 percent of the time intervals of the day for medium sections, as illustrated in Figure 7. Usually a low MAPE value is desirable for an algorithm. However, an algorithm with a low value of MAPE may occasionally yield a prediction with a large deviation. This is undesirable since it may divert passengers away from the bus stop and eventually cause them to miss the bus.

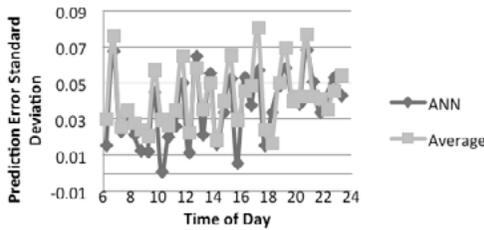
Therefore, it was important to define a second measure that could be used to detect this phenomenon. It examined the robustness of an algorithm such that its maximum deviation was within a certain range. Here, the robustness measure  $R_o$  is defined as:  $R_o = \max \{MAPE\}$  of a section at a certain time interval of the day. As can be seen from Figure 7, the maximum MAPE of the historical average approach is greater than the corresponding MAPE of the ANN. Hence, the proposed ANN is more robust than the historical average approach in terms of this measure.

It has been noted that standard deviation of prediction errors can also be used as a means of performance measure to make further comparison between the ANN and average models. For each section, prediction errors, i.e., the difference between the actual travel time and the forecast travel time, was calculated for each time interval. Figure 8 shows the standard deviation of the prediction errors for the sections discussed above over time of the day.

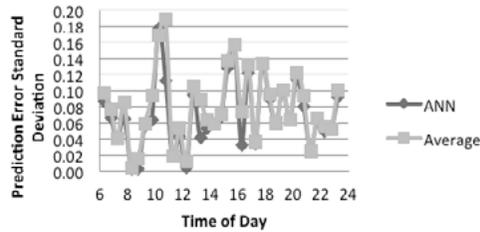
As can be seen from Figure 8, there was not much difference for the small and large sections. However, for the medium sections, the ANN appears to outperform the average model again, which reinforces our previous discussion. Therefore, it can be concluded that the ANN outperformed the average approach in terms of both prediction accuracy and robustness.



a) Short section (Section 1)



b) Medium section (Section 2)



c) Long section (Section 3)

**Figure 8. Standard deviation of prediction errors**

## Summary and Future Research

This paper presents an algorithm based on an artificial neural network using GPS data for predicting the travel time of transit vehicles between current bus location and any downstream bus stop under consideration. The predicted travel times between subsections of the route under consideration were compared with the measured data. The performance of the model was also tested and compared with a historical average approach, where the predicted travel time was taken to be the average of the travel times of previous buses that traveled between any

two stops under consideration. Prediction accuracy and robustness were used as performance measures. The overall precision measure determines the average deviation of the predicted travel time from the observed travel time. The robustness measure determines if an algorithm will occasionally give a prediction that is far off the actual arrival time.

The ANN outperformed the average approach in both performance measures. The standard deviation of the prediction errors reinforced this as well. The ANN model, which is trained off-line, enabled us to provide real-time travel time information at downstream stations with minimal error. The results obtained from the overall study are promising, and the proposed ANN model can be used to implement an APTS to predict the arrival time at bus stops in areas, even where there is undisciplined traffic flow. The implementation of this system will improve the reliability of the public transport system, thus attracting more travelers to buses and helping relieve congestion.

The study investigated the possibility of providing travel time information to transit users using departure and arrival time information at stops collected via GPS technology. Apart from prediction, creating an optimal bus schedule is also very important. Further study efforts may be directed towards incorporating schedule adherence information as an additional independent variable to improve the prediction model. Prediction with Kalman filtering algorithms in the absence of traffic data and comparing their results with the proposed model can also be conducted in the future. A user-interactive system may also need to be developed to provide the travel time information.

## **Acknowledgment**

The authors would like to thank Rápido Macaense and APB Prodata for the supply of travel time data for bus route LT11 in Macae, which was used for the case study. The data were provided during an original thesis study in 2010 by the first author.

## **References**

Chen, M., X. Liu, J. Xia, and S. Chien. 2004. A dynamic bus arrival time prediction model based on APC data. *Journal of Computer-Aided Civil and Infrastructure Engineering* 19(5): 364-376.

- Chien, S. I. J., Y. Ding, and C. Wei. 2002. Dynamic bus arrival time prediction with artificial neural networks. *Journal of Transportation Engineering* 128(4): 29-438.
- Dewan, K. K., and I. Ahmad, I. 2007. Carpooling: A step to reduce congestion (a case study of Delhi). *Engineering Letters* 14:1, International Association of Engineers. Retrieved March 2013 from [http://www.engineeringletters.com/issues\\_v14/issue\\_1/EL\\_14\\_1\\_12.pdf](http://www.engineeringletters.com/issues_v14/issue_1/EL_14_1_12.pdf).
- Farhan, A., A. Shalaby, and T. Sayed. 2002. Bus travel time prediction using GPS and APC. ASCE 7th International Conference on Applications of Advanced Technology in Transportation, Cambridge, Massachusetts.
- Hagan, M. T., H. B. Demuth, and M. Beale. 1996. *Neural Network Design*. PWS, Boston.
- Haykin, S. 1999. *Neural Networks: A Comprehensive Foundation*, 2nd Edition. Prentice-Hall, Upper Saddle River, NJ.
- Houghton, J., J. Reiners, and C. Lim. 2009. Intelligent transport system: How cities can improve mobility. *IBM Institute for Business Value*, U.S.A., Retrieved March 2013 from <ftp://ftp.software.ibm.com/common/ssi/pm/xb/n/gbe03232usen/GBE03232USEN.pdf>.
- Jeong, R., and L. R. Rilett. 2004. Bus arrival time prediction using artificial neural network model. IEEE Intelligent Transportation Systems Conference, Washington, DC: 988-993.
- Kalman R.E. 1960. A new approach to linear filtering and prediction problems. *Transactions of the ASME—Journal of Basic Engineering* 82 (Series D): 35–45.
- Karlik, B., and A. V. Olgac. 2010. Performance analysis of various activation functions in generalized MLP architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems* 1(4):111-122.
- Mathworks, Inc. 2011. MATLAB Version 6.1 Software.
- Patnaik, J., S. Chien, and A. Bladihas. 2004. Estimation of bus arrival times using APC data. *Journal of Public Transportation* 7(1): 1-20.
- Patricia, C. and C. Robert. 2005. A method for optimal division of data sets for use in neural networks. *Knowledge-Based Intelligent Information and Engineering Systems* 3684.

- Ramakrishna, Y., P. Ramakrishna, V. Lakshmanan, and R. Sivanandan. 2006. Bus travel time prediction using GPS Data. *Proceedings, Map India*. Retrieved March 2013 from [http://www.gisdevelopment.net/proceedings/mapindia/2006/student%20oral/mi06stu\\_84.htm](http://www.gisdevelopment.net/proceedings/mapindia/2006/student%20oral/mi06stu_84.htm).
- Rumelhart, D. E., G. E. Hilton, and R. J. Williams. 1986. Learning representations of back-propagation errors. *Nature (London)* 323: 533-536.
- Shalaby, A. and A. Farhan. 2003. Bus travel time prediction model for dynamic operations control and passenger information systems. CD-ROM, 82nd Annual Meeting of the Transportation Research Board, Washington, DC.
- Thomas, T., W. A. M. Weijermars, and E. C. Van Berkum. 2010. Predictions of urban volumes in single time series. *IEEE Transactions on Intelligent Transportation Systems* 11(1): 71-80.
- Vanajakshi, L., S. C. Subramanian, and R. Sivanandan. 2009. Travel time prediction under heterogeneous traffic conditions using global positioning system data from buses. *IET Intelligent Transport System* 3: 1-9.
- Yu, L., S. Wang, and K. K. Lai. 2006. An integrated data preparation scheme for neural network data analysis. *IEEE Transactions on Knowledge and Data Engineering* 18(2): 217-230.
- Zhang, G., B. E. Patuwo, and M.Y. Hu. 1998. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting* 14(1): 35-62.

## **About the Authors**

**ZEGEYE KEBEDE GURMU** ([zgurmu@patriots.uttyler.edu](mailto:zgurmu@patriots.uttyler.edu)) is a traffic engineer at Teague Nall & Perkins, Inc.

**DR. WEI (DAVID) FAN** ([wfan7@uncc.edu](mailto:wfan7@uncc.edu)) is currently an associate professor of Civil and Environmental Engineering Department in the University of North Carolina at Charlotte (UNCC). Prior to that, he worked as an associate professor at the University of Texas at Tyler and was a senior optimization developer in the R&D department in SAS Institute, Inc., after he obtained a Ph.D. in Civil Engineering from the University of Texas at Austin in 2004. His primary research interests include public transportation systems planning, transportation network modeling, traffic simulation and operations, and operations research (particularly optimization).