Received: October 1, 2019

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

IoT Based Bus Arrival Time Prediction Using Artificial Neural Network (ANN) for Smart Public Transport System (SPTS)

Jalaney Jabamony[1]* **Ganesh Ramaswamy Shanmugavel**[1]

[1]*Department of Electronics & Communication Engineering,*
Noorul Islam Centre for Higher Education, India
* Corresponding author's Email: jalaney316@gmail.com

Abstract: Advancement of the public transport system is important to modern society for reliable performance. Intelligent public transport system can utilize the time very effectively to give better performance to the society. Fast advancement in equipment, programming, and correspondence innovations has encouraged the rise of Internet-associated devices that give perceptions and information gathering from modern reality. By interfacing, an internet-enabled device with the public transport system leads to the intelligent public transport system. This paper proposed intelligent public transport with IoT enabled system to give an accurate prediction to the arrival time of the bus to the particular bus stops. Here, Artificial Neural Network (ANN) is used as a prediction algorithm and ANN is trained with different traffic parameters and environmental conditions. Parameters which are considered in the proposed system includes Distance(D), Waiting Time at Stops (WTS), Red signal Duration at Traffic Signal (RSD), Traffic Density (TD), Turning Density (TRD), Rush hours (RH), Weekends (WE), Weather conditions (WC), Number of passengers in the bus (NP), Public Holiday (PH), Road Type (RT). These parameters and the measured real-time arrival time of the bus in different stops for 10 days is used for training the ANN. This trained ANN is implemented on the server-side. In this paper, the performance of the proposed IOT-ANN-SPTS system is compared with the existing previous related works. From the performance analysis, it is shown that the proposed system produced less error (MAE=134.582, RMSE=197.738, MAPE=28.87) when compared to the other methods.

Keywords: Intelligent transportation system, Internet of things, Internet-connected sensory devices, Artificial neural network, Smart public transportation system, Public transport system.

1. Introduction

Buses are the most generally utilized public transportation in urban communities today. To give a high caliber of administration to the vehicle framework, diverse mechanical advancements are utilized. With the help of hardwires, internet-enabled devices, and data control used in this advanced technology enhances valuable people's safety, time and cash. The Internet-of-Things (IoT) gives extraordinary chances to enhance the existing open transport framework by installing keen innovation into the public transportation system. Time management in the public transport system must be adequate, so the passengers can use it as effectively. The model of SPTS requires information from all bus operators about the bus timing and bus cancellations. At a time, instant, it is important to use the prediction model to find the waiting time of the passengers to use the public transport system effectively.

The Internet of Things alludes to the consistently developing system of physical items that highlight an IP address for the internet network, and the correspondence that happens between these objects and other Internet-enabled devices and frameworks. Essentially, IoT is a system in which every single physical item is associated with the internet through system devices or switches and exchange information. Application of IoT includes Personal Home Automation System, Energy Management, Medical and Health Care, Transportation, Large-scale deployment. Integration of IoT to the public transport system increase the performance of a smart public transport system.

It is important to predict the arrival time of the public transport system to provide a reliable service to passengers. It allows passengers to plan their trips more reasonable and decreases their waiting time [4]. However, the precise prediction of bus arrival time is still challenging due to a number of stochastic factors (e.g., traffic congestion, weather conditions, and intersection delays).

The essential objective of the proposed method is to limit the waiting time of the traveler. The proposed strategy utilizes an Artificial Neural Network (ANN) to foresee the waiting time. Different direct and nonlinear course parameters considered to play out the forecast procedure. It utilizes historical information to train ANN. The ANN uses the current data from the server and also the historic data for the prediction. The remaining of the paper is structured as follows. In section 2, related work is discussed. The proposed methodology is explained briefly in section 3. Section 4 discusses data collection and Extraction. The result, discussion, and conclusion are given in sections 5 and 6 respectively.

2. Literature review

An intelligent Public transport system should be reliable to provide efficient service to the passengers. SPTS [1] using IOT shows that how IOT can be applied to PTS and present the navigational facilities for urban bus passengers. SPTS provides three novel information services for bus passengers: 1) Micro-navigation 2) Crowd-aware route recommendation and 3) Bus arrival time estimation. The solution in this work [2] involves using the existing internet-enabled devices on the bus (like the e-ticketing system) or a simple android tablet to capture the real-time location and send it to the servers. Accessing this location data from servers will be facilitated by Representational State Transfer (REST) APIs which users can access through the android application, SMS or web-portals. Enhanced particle filter algorithm [3] takes smartphone GPS positioning data [from multiple passengers in a single transit vehicle (e.g. bus)] as input data to track the position of the bus. This ‘crowdsourced’ data can then be utilized to calculate the vehicles’ positioning information with better accuracy using the developed enhanced particle filter algorithm. The paper [4] presents an IOT based sensing platform for smart cities called Protolyzing Lab. It is used to sense weather, environment, public transport and people flows in cities. In this, the data collected from the field (Cities) are stored in common backend using patio-temporal models to simplify sharing and joint analysis for the characterization of urban dynamics. This paper [5] presents an urban IoT system that is used to an ITS. It uses advanced and powerful communications (Wi-Fi, 3G/4G, and WiMax or GPRS). The IoT system supports intelligent traffic management, Driver management, and Parking management. A prediction model based on SVM [4], which used time, weather, road segment, running time of the current road segment and running time of the next link as the input parameters to predict the running time of the bus. An artificial neural network model based on site is used to predict bus arrival time [5]. Artificial neural network [6] has the ability of largescale parallel processing and distributed storage of information, and good self-learning ability. With the absolute superiority in prediction accuracy, it is a highly respected model of bus arrival time prediction. A Multi-index evaluation method based on Support Vector Machine (SVM) [7] is used for bus arrival time prediction. This method uses three new indexes including GPS coverage, release rate, and accuracy rate to evaluate the prediction service, and then uses SVM to train the model for multi-index evaluation. A real-time bus arrival prediction system [8] presents a model-based algorithm by providing the passenger with accurate information about bus arrival time. real-time data is automatically updated in this system with the help of passengers. Easy Come Easy Go (ECEG) [9] study presents an algorithm that uses real-time GPS data from the field and takes delays automatically into account for an accurate prediction of bus arrival time. ECEG system is primarily relying on powerful smartphones installed on a bus and independent from the bus operating companies. The strong performance of Kernel Regression algorithms [10] for the prediction of bus arrival times, which clearly outperform Linear Regression or K-Nearest Neighbour approaches. Bus arrival time prediction with GPS data in a real-time system consists of a bus arrival time estimation algorithm [11]. In this work, the SPTS is implemented with the IoT system to predict the arrival time of the bus. The algorithm used for the prediction of waiting time is ANN. IoT system is used to collect real-time data and sent it to the cloud or server. In the server, the waiting time is predicted with the current scenario of traffic conditions.

3. Proposed system

The bus arrival time prediction is an exceptionally complex issue, which may be affected by numerous parameters. The basic data used to predict bus arrival time including Distance(D), Waiting Time at Stops (WTS), Red signal Duration at Traffic Signal (RSD), Traffic Density (TD), Turning Density (TRD), Rush hours (RH), Weekends (WE), Weather conditions (WC) etc.

The primary function of the smartphone installed on the bus is to collect real-time GPS data including latitude, longitude, actual time, bus speed information and periodically transmit this information to the back-end server at every regular time. The back-end server receives the information from all smartphones that have been installed on the bus, and then provides the bus arrival time prediction using ANN. When travellers use the smartphone login system and send its GPS references and the route number desired to the back-end server, the back-end server will send back the arrival time of the bus wanted. By using this system, travellers can adjust their travel plans based on this information, and they can save their time and go to their destination as quickly as possible. The smartphone contributes the sensing information to the system. After the bus left the starting point, the data collection module starts to collect a sequence of GPS references. The collected data is transmitted to the server via cellular networks. The smartphone automatically collects data and transmits it without the help of the bus driver. Two stages are involved in this component. In order to bootstrap the system, it is necessary to survey the corresponding bus routes in the offline pre-processing stage. A basic database that associates particular bus routes to the bus stop with GPS references is created. It also requires the absolute physical location reference that helps to record the GPS references of an observed bus stop, which significantly reduces the initial construction overhead. As depicted in Figure. 1, a querying user queries the bus arrival time by sending the request to the backend server. The querying user indicates the interested bus route and bus stop to receive the predicted bus arrival time.

The proposed method uses an Artificial Neural Network (ANN) to predict the waiting time. Various linear and nonlinear route parameters considered to perform the prediction process. By using the proposed methodology, we can increase the accuracy of the system. The bus arrival time prediction is a very complex problem, which involves many factors influenced by random. Figure 2. Shows the detailed block diagram of the proposed system. When a passenger requests for the arrival time of a particular bus, he/she send a request to the server or cloud through the android app on their mobile phone. Here the connection between mobile phone and cloud is provided with IoT. The IoT sends the current location of the passenger to the cloud. The current location and traffic conditions of the requested bus are obtained through IoT. Then Fist historical data about the traffic condition is trained with ANN. Then real-time traffic data is passed to the cloud through IoT. The trained ANN then predicts the arrival time of the given bus to the user’s mobile phone. The basic data we used to predict bus arrival time including the following parameters.

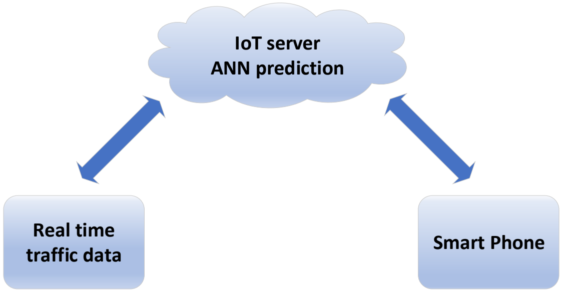


Figure.1 Basic block diagram of the proposed system

3.1 Parameters

The actual operation of the bus will be affected by many random factors, such as time, holidays, traffic congestion, traffic accidents [13], so it is very difficult to predict the arrival. Some of the factors that affect the speed of the public transport system are given below,

- Distance(D)
- Waiting Time at Stops (WTS)
- Red signal Duration at Traffic Signal (RSD)
- Traffic Density (TD)
- Turning Density (TRD)
- Rush hours (RH)
- Weekends (WE)
- Weather conditions (WC)
- Number of passengers on the bus (NP)
- Public Holiday (PH)
- Road Type (RT)
- Average Vehicle Speed limit (AVSL)

Distance between source to destination is one of the important parameters that affect the arrival time of the bus to the destination. The distance between the current bus location and querying passengers will be obtained by the GPS location of the bus and querying passengers. Traffic density is the number of vehicles occupying a given length of the given highway in a traffic lane. It is expressed as

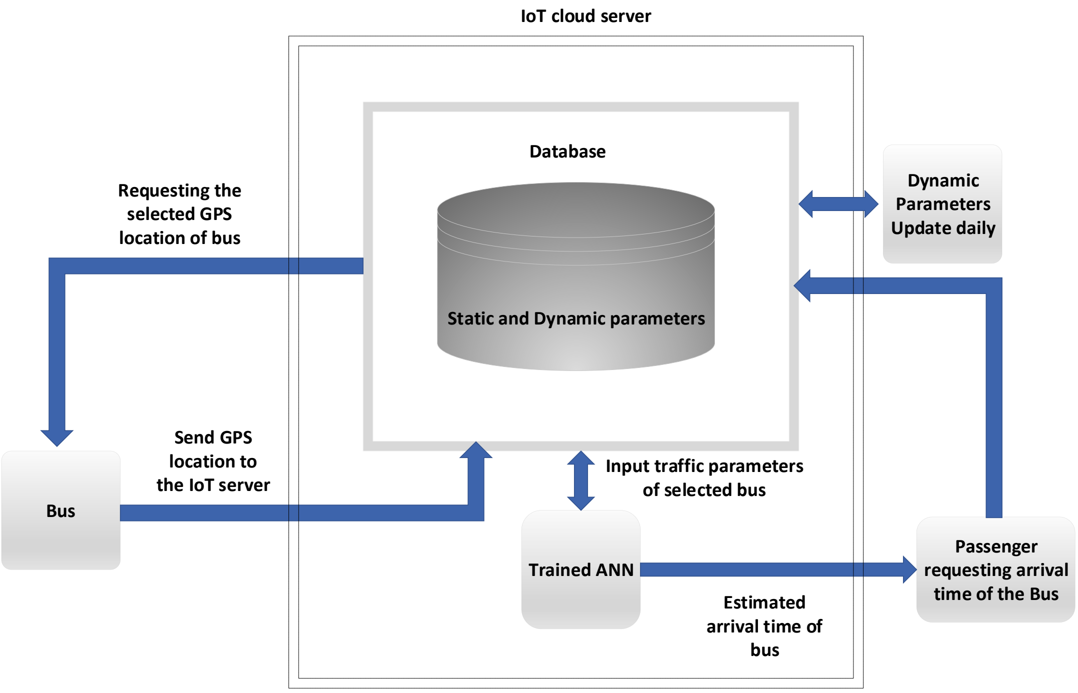

IoT cloud server
Database
Static and Dynamic parameters
Dynamic Parameters Update daily
Requesting the selected GPS location of bus
Send GPS location to the IoT server
Bus
Input traffic parameters of selected bus
Trained ANN
Passenger requesting arrival time of the Bus
Estimated arrival time of bus

Figure.2 Detailed block diagram of proposed IOT-ANN-SPTS

vehicle/mile or vehicle/kilometre. Traffic density is one of the important factors that increase the waiting time of the bus to reach the destination. Traffic density data is obtained from Google map. Rush hour is important factor during the time between morning 8 o'clock to evening 7 o'clock where the traffic density will vary. This time number of passengers in the bus is increased. So, this in turn increase the arrival time of bus to the destination point. Public holiday and week end also affects the waiting time of the bus. The speed of the bus is also affected during rainy days. Which in turn increase the arrival time of the bus. Road Type, Red signal Duration at Traffic Signal, Turning Density, Road Type, Average Vehicle Speed limit are some of the important factors that have an effect on the arrival time of the bus.

3.2 Mathematical modelling

Mathematical modelling is important to give the verification of theoretical details of the system to be designed. Representation each term in figure 3. is given by.

D - Distance from Source to Destination

TS_n -n^{th} Traffic Signal

TD_n -n^{th} Traffic Density at n^{th}Signal

S_n -n^{th} Stop

RSD_n -Red Signal Duration at n^{th}Traffic Signal

T_n - n^{th}Turn

WT_n - Waiting time at n^{th}Stop.

Distance(D) from source to destination can be obtained from the GPS location point. The expected time is calculated by

$$T = \frac{D}{ASL} \tag{1}$$

Waiting Time at Stops (WTS), Red signal Duration at Traffic Signal (RSD) are an important factor that affects the time, which must be added to the T_{mean} to obtained waiting time.

Total waiting time at stops are calculated by adding waiting time at each stop, which is given by

$$WTS_n = WTS_1 + WTS_2 + \cdots + WTS_n \tag{2}$$

TFPS = 8.333m/sec, Where

TFPS – Traffic Flow Per Second

Moving Time (MT) is given by

$$MT = TD/TFPS \tag{3}$$

$$WTTS = MT + RSD \tag{4}$$

Where, RSD – Red Signal Duration

$$WTTS_n = WTTS_1 + WTTS_2 + \cdots + WTTS_N \qquad (5)$$

Where, WTTS – Waiting Time at Traffic signal Signal-1

$$SR = AVSL\left[1 - \left(W_{trd} + W_{rh} + W_{we} + W_{wc} + W_{ph} + W_{np}\right)\right] \qquad (6)$$

$$Arrival\ Time = \left(\frac{Distance(D)}{Average\ SpeedOn\ Route(ASR)}\right)60 + WTS_n + WTTS_n$$

Where the values of weighting factors for each parameter such as Turning Density, Rush hours, Weekends, Weather conditions, Public Holiday respectively W_{trd}, W_{rh}, W_{we}, W_{wc}, W_{ph} are given by Table 1. Finally, estimated arrival time is calculated.

3.3 Arrival time prediction using Artificial neural network

In the proposed system ANN is used to predict the arrival time of the bus [13]. Artificial neural networks (ANN) or connectionist systems are computing systems that are inspired by, but not necessarily identical to, the biological neural networks that constitute animal brains. Such systems "learn" to perform tasks by considering examples, generally without being programmed with any task-specific rules [14].

ANN application areas ...

- Science and medicine: modelling, prediction, diagnosis, pattern recognition
- Marketing and Sales: analysis, classification, customer targeting
- Security: bomb, iceberg, fraud detection, etc.

Table 1. Maximum Weightage of Each parameter to calculate the average speed

Parameters	Representation	Maximum weightage (RT=1)	Maximum weightage (RT=2)
Weather conditions	W_{wc}	0.2	0.2
Rush hours	W_{rh}	0.15	0.15
Week ends	W_{we}	0.15	0.15
Public Holiday	W_{ph}	0.1	0.1
Turning Density	W_{trd}	0.05	0.05
Number of passengers in the bus	W_{np}	0.05	0.05

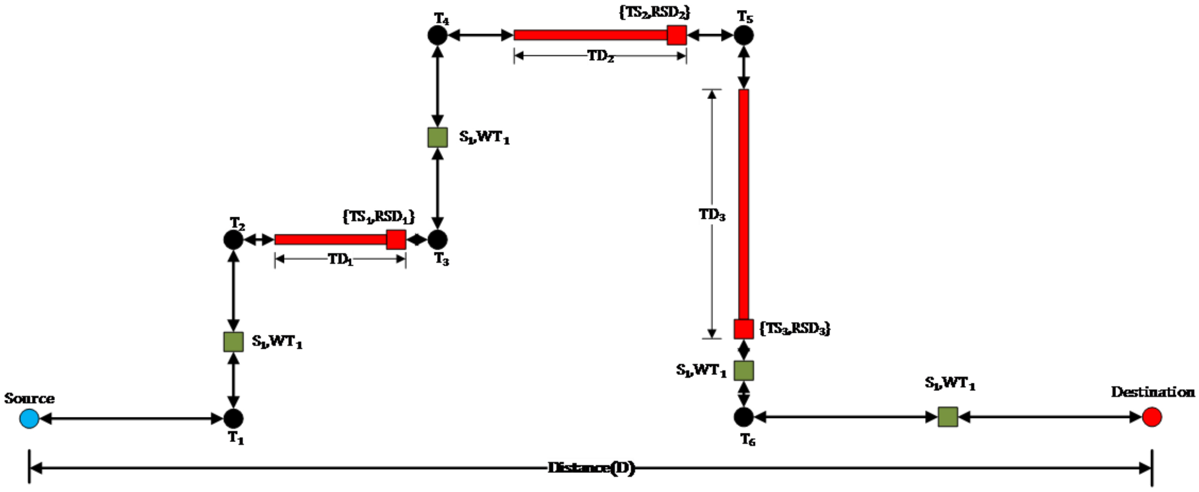


Figure.3 Path model for the proposed system

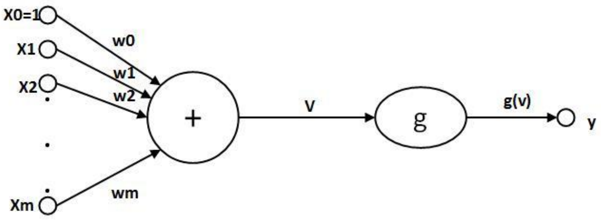


Figure.4 ANN model

ANN are relatively crude electronic networks of neurons based on the neural structure of the brain. The process records one at a time, and learn by comparing their prediction of the record (largely arbitrary) with the known actual record. The errors from the initial prediction of the first record are fed back to the network and used to modify the network's algorithm for the second iteration. These steps are repeated multiple times.

Artificial neural networks are moderately unrefined electronic networks of neurons dependent on the neural structure of the mind. Neurons are organized into the following layers: input, hidden, and output. The Layers are made up of a number of interconnected nodes that contain an activation function. Patterns are presented to the network via the input layer, which communicates to one or more hidden layers where the actual processing is done via a system of weighted connections. The final layer is the output layer, where there is one node for each class. Fig.4. shows the ANN model of the proposed system where {x0, x1, x2…xm} denotes the input, {w0, w1, w2…wm} denotes the weight assigned for every input, 'y' represents the output of the ANN model. A single sweep forward through the network results in the assignment of a value to each output node. The record is assigned to the class node with the highest value.

3.3.1. Training an artificial neural network

In the training phase, the correct class for each record is known (i.e., supervised training), and the output nodes can be assigned correct values -- 1 for the node corresponding to the correct class, and 0 for the others. Results have been found using values of 0.9 and 0.1, respectively. As a result, it is possible to compare the network's calculated values for the output nodes to these correct values and calculate an error term for each node. These error terms are then used to adjust the weights in the hidden layers so that the next time around the output values will be closer to the correct values.

3.3.2. The iterative learning process

A key feature of neural networks is an iterative learning process in which records (rows) are presented to the network one at a time, and the weights associated with the input values are adjusted each time. After all, cases are presented, the process often starts over again. During this learning phase, the network trains by adjusting the weights to predict the correct class label of input samples. The advantages of neural networks include their high tolerance to noisy data, as well as their ability to classify patterns on which they have not been trained. The most popular neural network algorithm is the back-propagation algorithm. Once a network has been structured for a particular application, that network is ready to be trained. To start this process, the initial weights are chosen randomly. Next, the training begins.

The network processes the records in the training data one at a time using the weights and functions in the hidden layers then compares the resulting outputs against the desired outputs. Errors are then propagated back through the system, causing the system to adjust the weights for the next record. This process occurs again as the weights are continually tweaked. During the training of a network, the same set of data is processed many times as the connection weights are continually refined.

Note that some networks never learn. This could be because the input data does not contain the specific information from which the desired output is derived. Networks also will not converge if there is not enough data to enable complete learning. Ideally, there should be enough data available to create a Validation Set.

3.3.3. Feedforward, back-propagation

The typical back-propagation network has an input layer, an output layer, and at least one hidden layer. Theoretically, there is no limit on the number of hidden layers, but typically there are just one or two. Some studies have shown that the total number of layers needed to solve problems of any complexity is five (one input layer, three hidden layers, and an output layer). Each layer is fully connected to the succeeding layer.

The training process normally uses some variant of the Delta Rule, which starts with the calculated difference between the actual outputs and the desired outputs. Using this error, connection weights are increased in proportion to the error times, which are a scaling factor for global accuracy. This means that the inputs, the output, and the desired output all must be present at the same processing element. The most complex part of this algorithm is determining which input contributed the most to incorrect output and how to modify the input to correct the error. (An inactive node would not contribute to the error and would have no need to change its weights.) To solve this problem, training inputs are applied to the input layer of the network and desired outputs are compared at the output layer.

During the learning process, a forward sweep is made through the network, and the output of each element is computed layer by layer. The difference between the output of the final layer and the desired output is back-propagated to the previous layer(s), usually modified by the derivative of the transfer function. The connection weights are normally adjusted using the Delta Rule. This process proceeds for the previous layer(s) until the input layer is reached.

3.3.4. Structuring the network

The number of layers and the number of processing elements per layer are important decisions. To a feedforward, these parameters back-propagation topology, are also the most ethereal - they are the art of the network designer. There is no quantifiable, best answer to the layout of the network for any particular application. There are only three general rules picked up over time and followed by most researchers and engineers applying this architecture to their problems.

Rule One: As the complexity in the relationship between the input data and the desired output increases, the number of the processing elements in the hidden layer should also increase.

Rule Two: If the process being modelled is separable into multiple stages, then additional hidden layer(s) may be required. If the process is not separable into stages, then additional layers may simply enable the memorization of the Training Set and not a true general solution.

Rule Three: The amount of training data available sets an upper bound for the number of processing elements in the hidden layer(s). To calculate this upper bound, use the number of cases in the Training Set and divide that number by the sum of the number of nodes in the input and output layers in the network. Then divide that result again by a scaling factor between five and ten. Larger scaling factors are used for relatively less noisy data. If too many artificial neurons are used, the Training Set will be memorized, not generalized, and the network will be useless on new data sets.

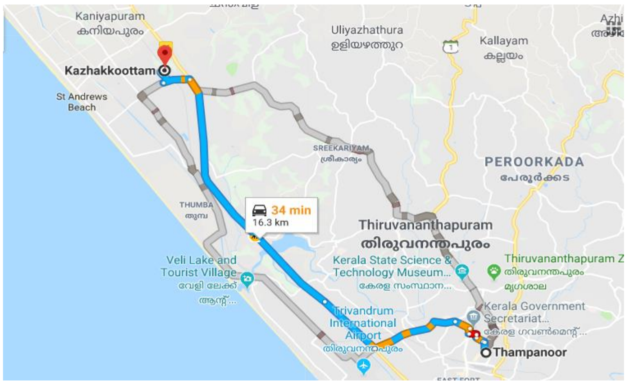


Figure.5 Google Map for selected source to destination in the proposed system

4. Data collection and extraction

The information prerequisite for such an investigation ought to in perfect between the selected route for several days. In any case, under Indian activity conditions, such an information document isn't accessible and consequently one needs to rely upon restricted information that is gathered physically. So, there is a lack of similar studies. The selected route for the proposed system is the public bus route in Trivandrum city. The selected route connects tampanoor bus station to kazhakootum bus station. In between source to destination, there are lots of schools, colleges, and many companies. This is one of the longest and important routes within the city and is approximately 16.4 km long with a travel time of about 30-34 min. There are 21 bus stops (including the terminal stops in this route). Figure. 5. Traffic density information, weather conditions are obtained from the Google map. Other information [15] such as Public Holiday, Rush hours, Public Holiday, Road Type, Average Vehicle Speed limit, Number of passengers in the bus, Waiting Time at Stops, Red signal Duration at Traffic Signal are manually obtained for 20 days of observations. These parameters and real arrival time measured are used to train the ANN.

5. Result and discussion

Data collected for 20 days are given to the neural network. The figure 6: shows the neural networks used in the proposed system. ANN is trained with 100 hidden layers to produce error-free output. Figure 6. Shows the structure of ANN used in the proposed system.

Table 2 and 3 gives the estimated arrival time of the selected bus in a given route. From this table, it is shown that ANN has higher performance when

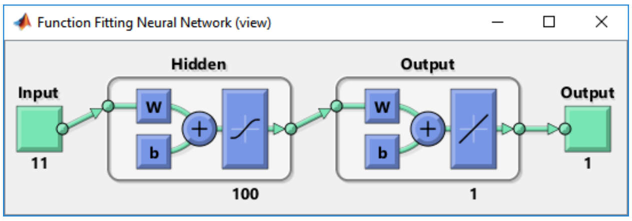


Figure .6 ANN architecture of the proposed system

compared to the KNN [17]. The estimated value obtained by ANN is almost equal when compared to the observed arrival time.

Fig. 7 shows the estimated arrival time in public holidays and working days. From this graph, it is shown that in the working days the estimated arrival time is high when compared to that on holiday. This is because of the reason in the public holiday the number of vehicles, passengers, and traffic density in the road is less.

Fig. 8 shows the estimated arrival time when the number of passengers on the bus changed during public holiday and working day. From this graph, it is shown that ANN accurately differentiates the public holiday and workday traffic conditions.

Fig. 10 shows the simulation results of both KNN and the proposed system. The proposed system is simulated with MATLAB 2017b. Figure 11 shows the error graph of both KNN and the proposed ANN structure. This shows that the proposed ANN presents fewer errors when compared to KNN. This is because the proposed system considered all traffic and environment parameters which affect the smooth traffic flow conditions. Figures 9 and 10 show the arrival time estimation when traffic density and red signal duration changed are changed during the working day and public holiday. From this graph, it is shown that traffic density and red signal duration explicitly affect the arrival time of the bus.

Table 2. comparison for actual and predicted time for KNN [17] prediction

Distance (km)	1	5	10	15	20	25	30
Observed arrival time(minutes)	98.60	101.09	103.65	106.20	108.75	111.31	113.86
Estimated arrival time (minutes)	116.23	116.23	116.23	113.79	113.79	113.79	113.79

Table 3. Comparison for actual and predicted time for ANN

Distance (km)	1	5	10	15	20	25	30
Observed arrival time(minutes)	98.60	101.09	103.65	106.20	108.75	111.31	113.86
Estimated arrival time (minutes)	98.53	101.08	103.65	106.213	108.76	111.31	113.86

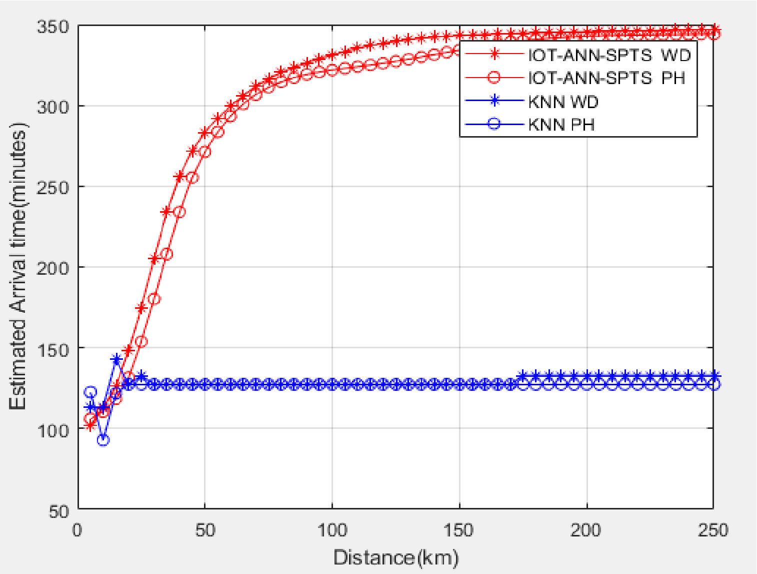


Figure.7 Output Estimated arrival time when the distance is changed during working and public holiday

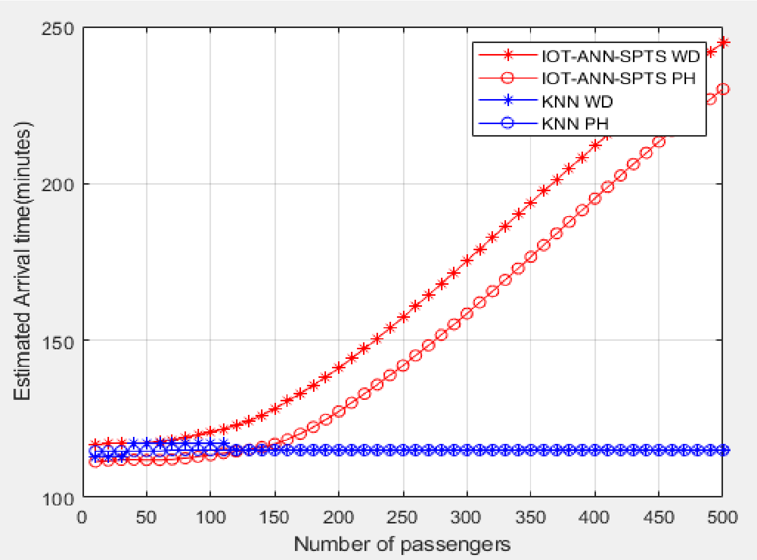


Figure .8 Output Estimated arrival time when number passengers are changed during working and public holiday

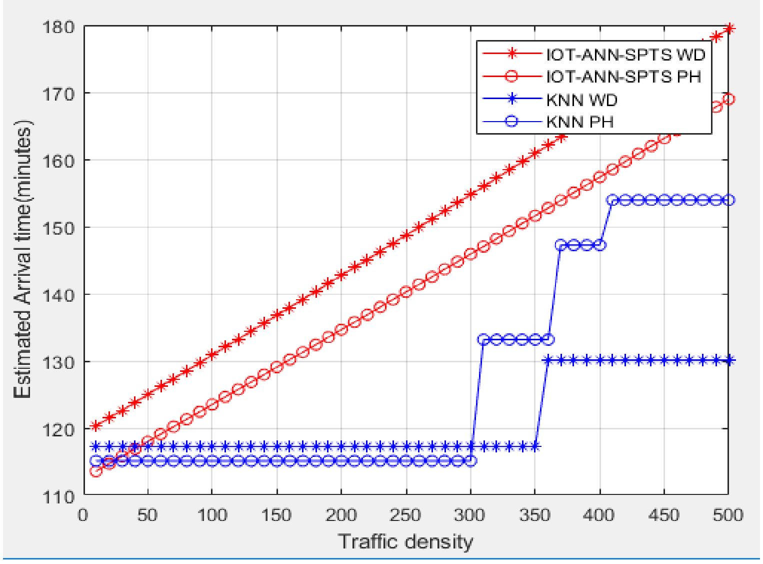


Figure.9 Output estimated arrival time when traffic density is changed during working and public holiday

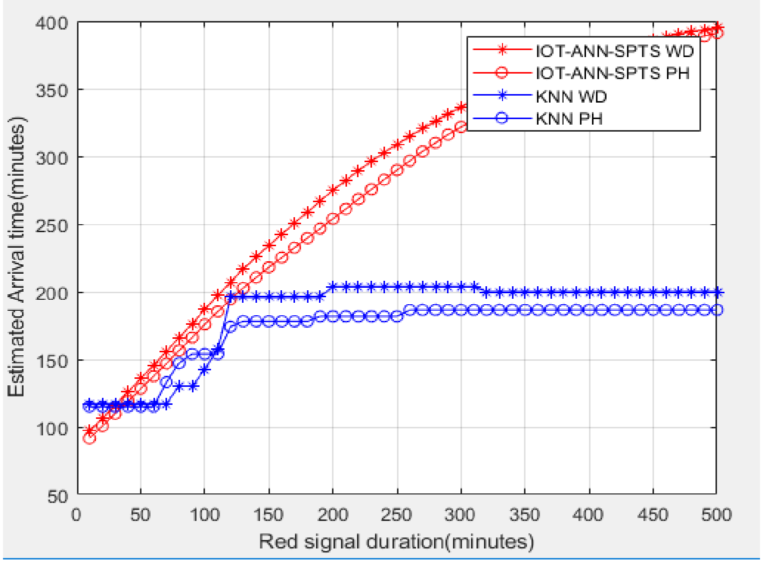
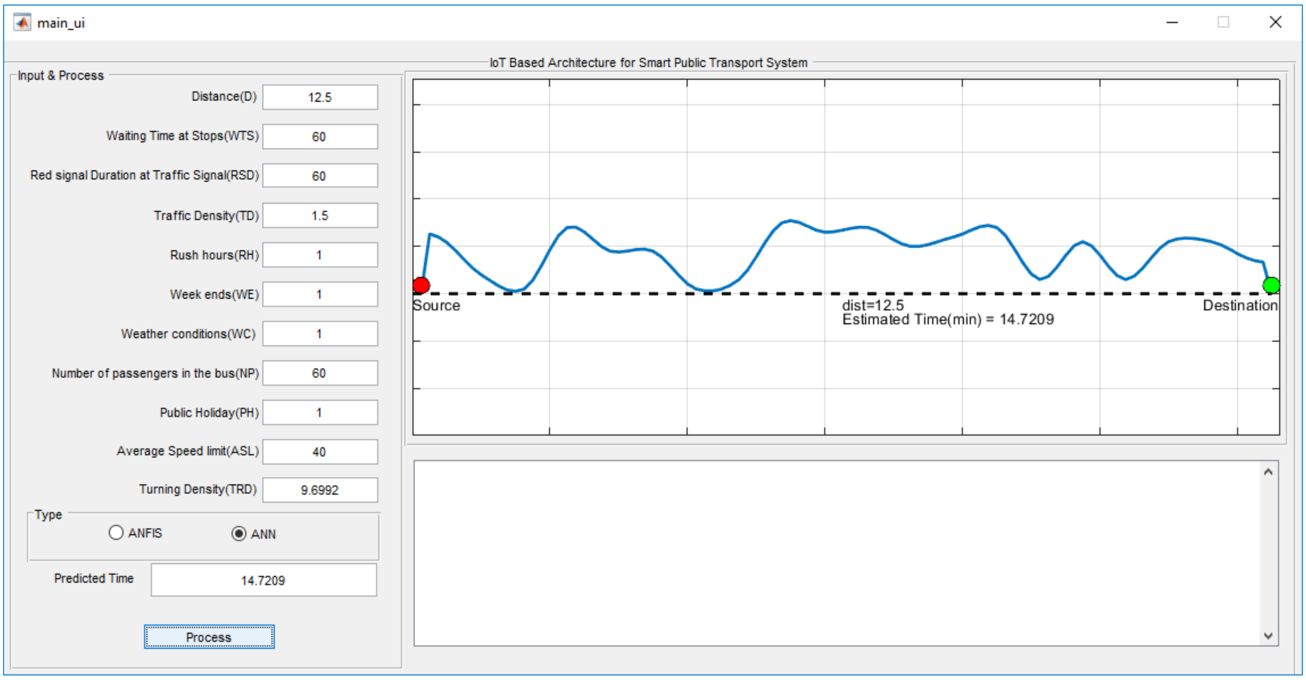


Figure.10 Output Estimated arrival time when the red signal duration is changed during working and public holiday



main_ui

Input & Process

Distance(D)	12.5
Waiting Time at Stops(WTS)	60
Red signal Duration at Traffic Signal(RSD)	60
Traffic Density(TD)	1.5
Rush hours(RH)	1
Week ends(WE)	1
Weather conditions(WC)	1
Number of passengers in the bus(NP)	60
Public Holiday(PH)	1
Average Speed limit(ASL)	40
Turning Density(TRD)	9.6992

Type ○ ANFIS ● ANN

Predicted Time 14.7209

Process
IoT Based Architecture for Smart Public Transport System
Source
dist=12.5
Estimated Time(min) = 14.7209
Destination

Figure.11 Simulation result of the proposed system

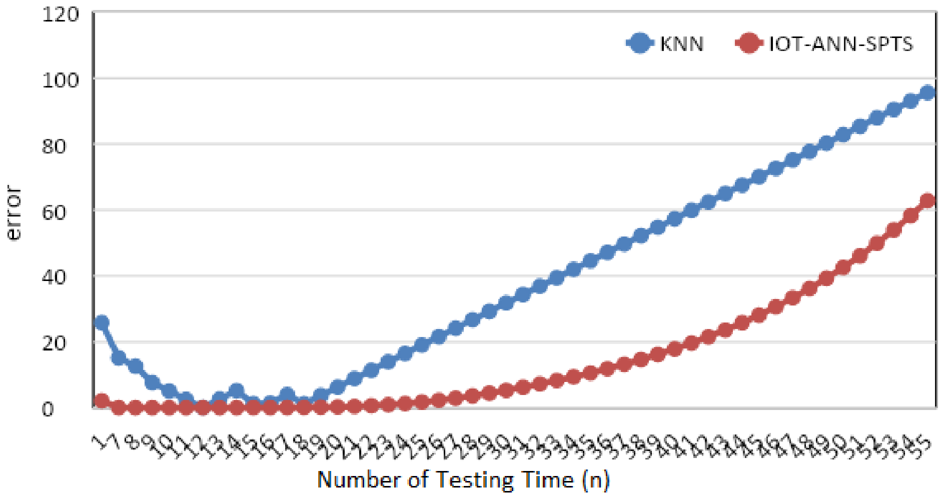


Figure .12 Error graph between the proposed system and KNN system

5.1 Model performance measures

The performance of the models in term of accuracy is evaluated by these three measures: mean absolute error (MAE; sec), root mean square error (RMSE; sec), mean absolute percentage error (MAPE; %) [16]. They are calculated as follows:

$$MAE = \frac{\sum\left|t^{actual} - t^{predict}\right|}{N} \qquad (8)$$

$$RMSE = \sqrt{\frac{\sum(t^{actual} - t^{predict})^2}{N}} \qquad (9)$$

$$MAPE = \frac{1}{N}\sum\frac{\left|t^{actual} - t^{predict}\right|}{t^{actual}} \cdot 100\% \qquad (10)$$

where t^{actual} is the actual observed bus travel time, $t^{predict}$ is the predicted bus travel time, and N is the number of testing times. The reason that we use all these three measures is their different

Table 4. Performance table

Parameters	*MAE*	*RMSE*	*MAPE*
IOT-ANN-SPTS	276.452	331.868	58.224
ANN	134.582	177.738	28.871

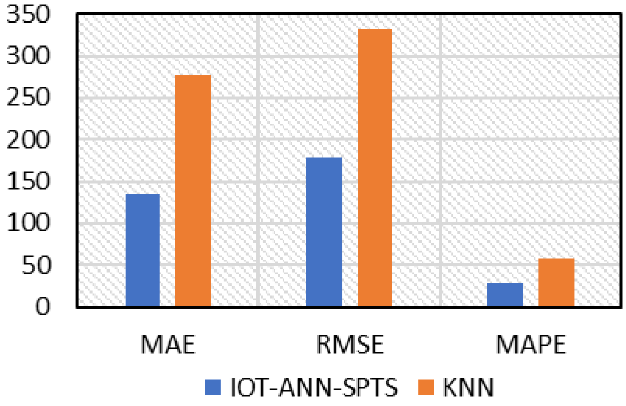


Figure.13 Performance graph

responses to the error. In general, if we want to know how much average error of the prediction, the magnitude of this error can be told by MAE. If the magnitude of the error is significant, RMSE is a measure that is sensitive to the error magnitude. Lastly, MAPE is easy to read. It can tell us how big the 10-seconds error is, in a percentage of the actual travel time.

From Table 4 and Fig. 13, it is shown that the proposed IOT-ANN-SPTS system has fewer errors when compared to the KNN.

6. Conclusion

IoT based Bus Arrival Time Prediction for SPTS is implemented using an Artificial neural network-based predictor. This paper investigated an approach to improve the performance of transport travel time prediction at any distance along with the route. To assess the performance of the model, transports' movement data was gathered from Tampanoor (Trivandrum) bus stop to Kazhakootum bus stop, India for more than 10 days. The proposed method gives a highly accurate arrival time of the bus in a particular bus station under different traffic conditions. Proposed IOT-ANN-SPTS is compared with another state of art method from that it is shown that the proposed IOT-ANN-SPTS method accurately gives an estimate of the arrival time of the bus when compared to KNN. From the performance analysis, it is shown that the proposed system produced less error (MAE=134.582, RMSE=197.738, MAPE=28.87) when compared to the other methods.

References

[1] S. Apsara, G. A. Rashmi, K. V. Mohan Kumar, L. Anitha, and B. J. yothi, “Smart Public Transport System Using Internet-Of-Things”, *International Journal of Advance Research, Ideas and Innovations in Technology*, Vol.3, No.2, pp. 1163-1170, 2017.

[2] L. Jay, R. Dani, S. Sontakke, and R. Adhao, “Scalable tracking system for public buses using IoT technologies”, In: *Proc. of International Conf. on Emerging Trends & Innovation in ICT (ICEI)*, pp. 104-109, 2017.

[3] P. Mukheja, N. R. Velaga, and R.B. Sharmila, “Smartphone-based crowdsourcing for position estimation of public transport vehicles”, *International Journal of IET Intelligent Transport Systems*, Vol. 11, No. 9, pp.588-595, 2017

[4] Y. Bin, Y. Zhongzhen, and Y. Baozhen, “Bus arrival time prediction using support vector machines”, *Journal of Intelligent Transportation Systems*, Vol. 10, No. 4, pp.151-158, 2006.

[5] S. I. Chien, Y. Ding, and C. Wei, “Dynamic bus arrival time prediction with artificial neural networks”, *Journal of Transportation Engineering*, Vol. 128, No. 5, pp. 429-438, 2002.

[6] Y. Juan-li, X. Mei, W. Fu-lin, W. Ji-quan, and L. Hui, “Research on time series forecasting based on BP neural network”, *Mathematics in Practice and Theory*, 2013.

[7] H. Zhiying, Y. Haitao, D. Yong, and J. Wang, “SVM based multi-index evaluation for bus arrival time prediction”, In: *Proc. of International Conference on ICT Convergence (ICTC)*, pp. 86-90, 2013.

[8] R.P.S. Padmanaban, K. Divakar, L. Vanajakshi, and S. C. Subramanian, “Development of a real-time bus arrival prediction system for Indian traffic conditions”, *International Journal of IET Intelligent Transport Systems*, Vol. 4, No. 3, pp.189-200, 2010.

[9] Y. Bin, H. K. L. William, and M. L Tam, “Bus arrival time prediction at bus stop with multiple routes”, *Transportation Research Part C: Emerging Technologies*, Vol. 19, No. 6, pp. 1157-1170, 2011.

[10] M. Sinn and J. W. Yoon, “Predicting arrival times of buses using real-time GPS measurements”, In: *Proc. of International Conf. on Intelligent Transportation Systems*, pp. 16-19, 2012.

[11] W. H. Lin and J. Zeng, “Experimental study of real-time bus arrival time prediction with GPS data”, *Transportation Research Record: Journal of the Transportation Research Board* 1666, pp.101-109, 1999

[12] L. Jinglin, J. Gao, Y. Yang, and H. Wei, “Bus arrival time prediction based on mixed model”, *International Journal of China Communications*, Vol. 14, No. 5 pp.38-47, 2017.

[13] R. Jeong and R. Rilett, “Bus arrival time prediction using artificial neural network model”, In: *Proc. of International Conf. on Intelligent Transportation Systems*, pp.988-993, 2004.

[14] W. G. Baxt, “Application of artificial neural networks to clinical medicine”, *The Lancet*, Vol. 346, No. 8983, pp.1135-1138, 1995.

[15] Q. Zhang, Y. Zhang, and L. Jingyi, “Easy Come EasyGo: Predicting bus arrival time with smart phone”, In: *Proc. of International Conf. On Frontier of Computer Science and Technology (FCST)*, pp. 268-273. 2015.

[16] W. Treethidtaphat, W. Pattara-Atikom, and S. Khaimook, “Bus arrival time prediction at any distance of bus route using deep neural network model”, In: *Proc. of International Conf. on Intelligent Transportation Systems (ITSC)*, pp. 988-992, 2017.

[17] X. Jianxia and S. Chen, “Bus dwell time prediction based on KNN”, *International Journal of Procedia Engineering*, Vol. 137, pp. 283-288, 2016.