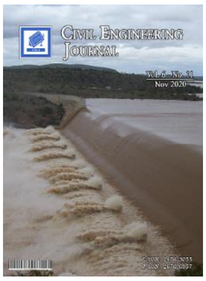
Available online at www.CivileJournal.org

Civil Engineering Journal

Vol. 6, No. 11, November, 2020



Predicting Travel Times of Bus Transit in Washington, D.C. Using Artificial Neural Networks

Stephen Arhin [a*], Babin Manandhar [b], Hamdiat Baba-Adam [a]

[a] *Associate Professor, Howard University Transportation Research Center, Washington, D.C., United States.*

[b] *Research Assistant, Howard University Transportation Research Center, Washington, D.C., United States.*

Received 31 July 2020; Accepted 17 October 2020

Abstract

This study aimed to develop travel time prediction models for transit buses to assist decision-makers improve service quality and patronage. Six-months' worth of Automatic Vehicle Location and Automatic Passenger Counting data for six Washington Metropolitan Area Transit Authority bus routes operating in Washington, D.C. was used for this study. Artificial Neural Network (ANN) models were developed for predicting travel times of buses for different peak periods. The analysis included variables such as length of route between stops, average dwell time and number of intersections between bus stops amongst others. Quasi-Newton algorithm was used to train the data to obtain the ideal number of perceptron layers that generated the least amount of error for all peak models. Comparison of the Normalized Squared Errors generated during the training process was done to evaluate the models. Travel time equations for buses were obtained for different peaks using ANN. The results indicate that the prediction models can effectively predict bus travel times on selected routes during different peaks of the day with minimal percentage errors. These prediction models can be adapted by transit agencies to provide patrons with more accurate travel time information at bus stops or online.

Keywords: Travel Time; Artificial Neural Network; Quasi-Newton Algorithm; Bus Transit.

1. Introduction

Washington, D.C. is ranked second among cities in terms of highest public transit commuters in the United States with approximately 9% of the working population using the Washington Metropolitan Area Transit Authority (WMATA) Metrobuses to commute [1]. The Metrobus in D.C. is the fifth largest bus system in the United States. It has over 1,450 buses and services approximately 350 routes across the D.C., Maryland and Virginia area [2]. The buses serve 11,129 stops, including 2,554 stops with bus shelters [3].

The accurate prediction of travel time is necessary to enable public transit agencies to provide patrons with efficient transit service and for them to effectively plan their commute or travel in the region. Transit agencies are continuously evaluating best practices available to improve reliability of their services. The use of technology, particularly in bus transit, has been critical for this purpose. This includes the use of Automatic Vehicle Location (AVL) technology, which has been instrumental in the tracking of buses in real-time. Automatic Passenger Counters (APC) installed on buses count the number of passengers alighting and boarding at each bus stop which helps in the computation of the total number of patrons onboard.

[*] Corresponding author: saarhin@howard.edu

http://dx.doi.org/10.28991/cej-2020-03091615

The use of public transportation instead of personal vehicles has been encouraged as one of the solutions to traffic congestion. A study conducted by Abdulrazzaq et al. (2020) in Kanjang City, Malaysia indicated that travel time and travel distance coupled with schedule accuracy, fare reductions and increased accessibility significantly influenced riders decision to use public transport [4]. Obtaining accurate travel times of the metrobuses is an important task for transit authorities to provide reliable service to its patrons. A study conducted in 2013 found that transit buses in DC had an overall on-time performance of approximately 75%. The scheduled arrival times and actual arrival times had mean deviations ranging between 1.99 and 5.03 minutes [5]. The cumulative deviations in arrival times can negatively affect patron's perception of transit reliability. It is therefore important that travel time prediction models are developed to provide more accurate information to patrons based on pertinent factors that affect travel and arrival times.

There are several bus arrival prediction models that have been developed using different techniques such as Historical average models, Regression Models, Kalman Filter models and Artificial Neural Network (ANN) models based on several different variables [6]. Significant variables used in the various bus arrival prediction models include time of the day, vehicle arrival/departure, speed, distance, passengers boarding/alighting, en-route traffic conditions, etc. Ranjitkar et al. (2019) conducted a study that introduced 10 independent variables (factors) in seven (7) different models to compare their accuracies in predicting bus travel time using AVL and APC data. The developed models used included multivariate linear regression, ANN, decision tree, gene expression programming models, among others. The study concluded that ANN model performed best in comparison to the other models and the distance between two stops was the most significant variable for all models [7].

Machine Learning models such as ANNs provide a much more effective alternative and better accuracies in comparison to the conventional models [8]. ANNs are mathematical models that are inspired by the biological neural networks in the human brain. The effectiveness of ANN is based on its ability to approximate both linear and nonlinear functions to a required degree of accuracy using a learning algorithm, and to build ''piece-wise'' approximations of the functions.

Jeong and Rilett (2004) developed historical data-based model, regression and ANN models and compared their performance using AVL data to predict bus arrival time in Houston, Texas. The prediction of bus arrival time was based on dwell time at stops and traffic congestion. The ANN model had the lowest Mean Absolute Percentage Error (MAPE) of about 54.24% and 48.61% when compared to the historical data-based and the regression models. The results from the study indicated that in terms of prediction accuracy, the ANN models outperformed the historical data model and regression models [8]. A study conducted by Chien et al. (2002) developed two ANN predictive models that were trained by link-based and stop-based data respectively to accurately predict bus arrival times in an urban road network. Though each model performed better under different scenarios, the study concluded that a hybrid ANN model which integrated both the link-based and stop-based models would further improve the accuracy of bus arrival time predictions [9].

Yu et al. (2017) developed accelerated time survival and linear regression models to estimate both travel times and the level of uncertainty associated with these predictions (travel time variance) based on a headway bus route data from Pennsylvania State University-University Park Campus. The results indicated that though the bus travel times prediction accuracies from both models were similar, the accelerated time survival models performed better with smaller uncertainties of 76% and reduced travel time prediction variation of 12% [10]. Treethidtaphat et al. (2017) designed a bus arrival prediction model using Deep Neural Network (DNN) based on GPS data from a public transportation bus line in Bangkok, Thailand. The results determined that the DNN model provided better bus arrival prediction times by up to 55% when compared an OLS model and the bus line's current prediction model [11].

Chen (2018) proposed an Arrival Time Prediction Model (ATPM) for passenger and/ or tourism systems using Hsinchu and Yosemite Bus systems in Taiwan as case study. Three months' worth of data for 14 highway routes and 40 urban roads were used in the ATPM based on Random Neural Networks (RNN). From the results, the ATPM produced better accuracies when compared to the conventional ANN models. A smart bus system designed using the proposed ATPM produced accuracies of 94.75% for highways and 78.22% for urban roads in providing travel time information to agency and riders [12]. A study by Yu et al. (2018) proposed and compared bus travel time prediction models based on Random Forests Near Neighbor (RFNN), Linear Regression (LR), K-Nearest Neighbor (KNN), Support Vector Machines (SVM) and classic Random Forests (RF). AVL data from two bus routes in Shenyang, China with comparable traffic situations were used. The results determined that though the RFNN bus travel time prediction model had the longest computation times, it generated better accuracies in the MAE, MAPE and RMSE [13].

Petersen et al. (2019) designed and implemented a multi-output and multi-time-step Deep Neural Network model which blends convolutional and Long Short-Term Memory (LSTM) layers to form a hybrid model (ConvLSTM). The model used six months' worth of urban bus transportation data provided by Movia Public Transport Authority, Copenhagen, comprising of AVL and GPS data. Results from the study showed that the ConvLSTM model performed better across all peaks with a MAPE of 4.04% in the morning peak and 5.61% in the afternoon peak when compared to the Historical Average model, the current bus line's prediction model, the LSTM model and Google maps traffic models. In addition, the ConvLSTM model could predict travel times for multiple links and multiple time-steps ahead [14].

Despite the studies and models developed, none has been developed specifically to predict bus travel time using ANN considering the unique nature of traffic patterns in the DC area. This research aimed at developing ANN models to predict travel time of transit buses in Washington, DC using AVL and APC data. The models will enable public transit agencies to provide more accurate travel time information to patrons to improve reliability and consequently increase bus ridership. This report presents the findings of the case study conducted to predict bus transit travel times in Washington, DC with ANN using AVL and APC data. The materials and methods section provides the description of the study site along with the selection process of bus routes. This section also provides the process of obtaining the data and filtering it based on relevant independent variables to generate data sets per peak period required for the analysis. The data analysis will constitute of a descriptive statistics section along with the ANN model development to find bus travel times. The training strategy and the optimization algorithm to perform the neural network analysis are described in this section. The results section presents the findings of the neural network training for the different case scenarios (varying number of perceptron layers). The initial and final training and selection errors are compared for every iteration to compare the highest percentage change in error. Furthermore, the Normalized Squared Errors of the testing datasets are compared to determine the accuracy and reliability of the training strategies and the travel time equations. A summary of the study is presented in the conclusion section.

2. Materials and Methods

This section provides the steps that were followed for the collection of data required for the analysis as shown in Figure 1.

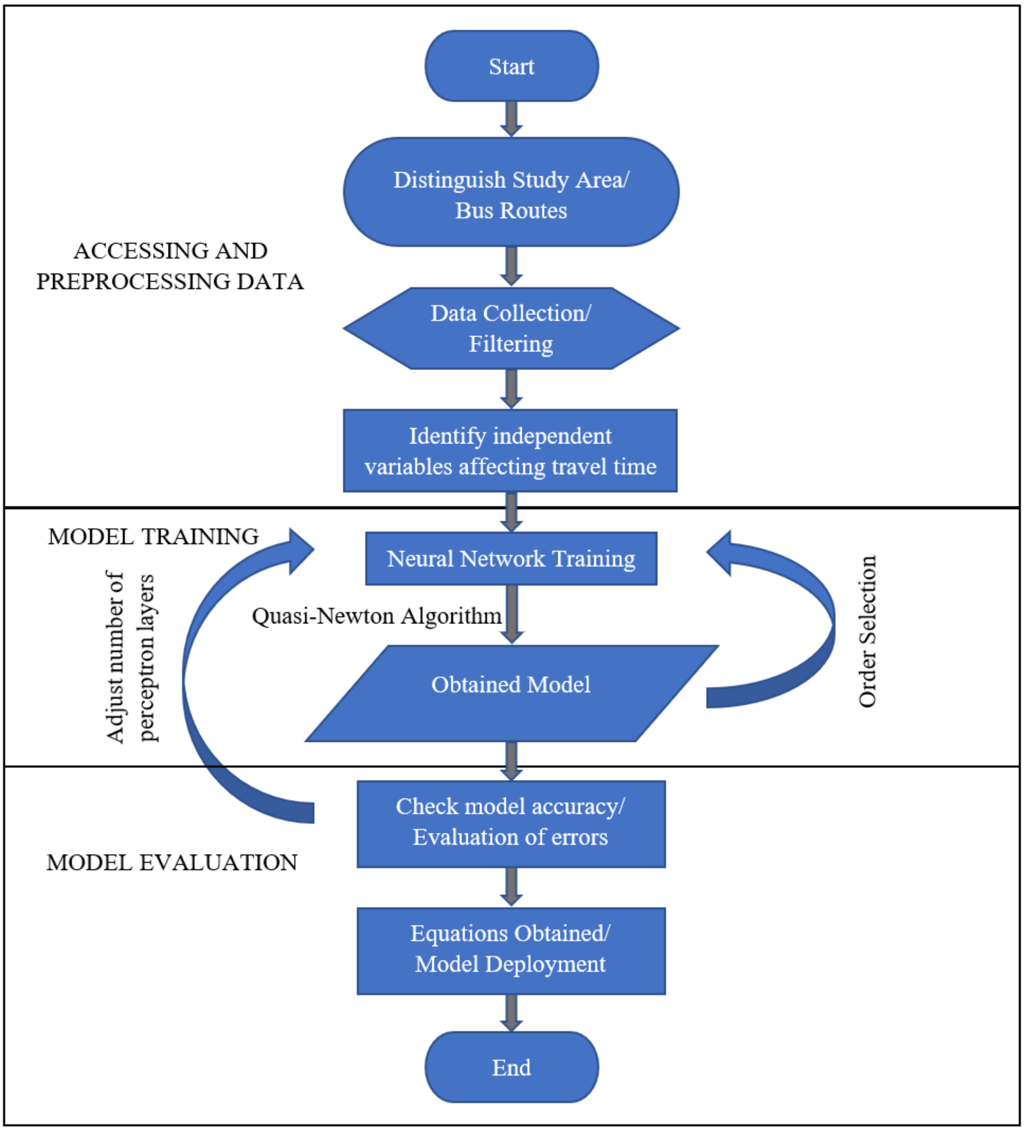

Start
Distinguish Study Area/ Bus Routes
ACCESSING AND PREPROCESSING DATA
Data Collection/ Filtering
Identify independent variables affecting travel time
MODEL TRAINING
Neural Network Training
Adjust number of perceptron layers
Quasi-Newton Algorithm
Order Selection
Obtained Model
Check model accuracy/ Evaluation of errors
MODEL EVALUATION
Equations Obtained/ Model Deployment
End

Figure 1. Flowchart of the study process

Figure 1 presents a flowchart of the methodology that was followed to obtain travel time equations for buses using neural network training.

2.1. Site Description and Selection of Bus Routes

The Washington Metropolitan Area Transit Authority (WMATA) oversees the operations of Metrobus service in Washington, DC. The city is divided into four (unequal) quadrants: Northwest (NW), Northeast (NE), Southwest (SW), and Southeast (SE). As of July 2018, the population of Washington, DC was approximately 702,455 with an annual growth rate of approximately 1.4% [15]. WMATA has a bus fleet of approximately 1,600 buses that operate on 325 routes in Washington, DC, in portions of Maryland, and Northern Virginia, covering a total land area of about 1,500 square miles. Metrobuses operate 24 hours a day, 7 days a week and make more than 400,000 trips each week day. Of the total number of bus stops, 2,556 (22.2%) have shelters, while the remainder do not [16]. Figure 2 represents a location map of Washington, DC (shown in red) to the left and a road map of Washington, DC to the right [17].

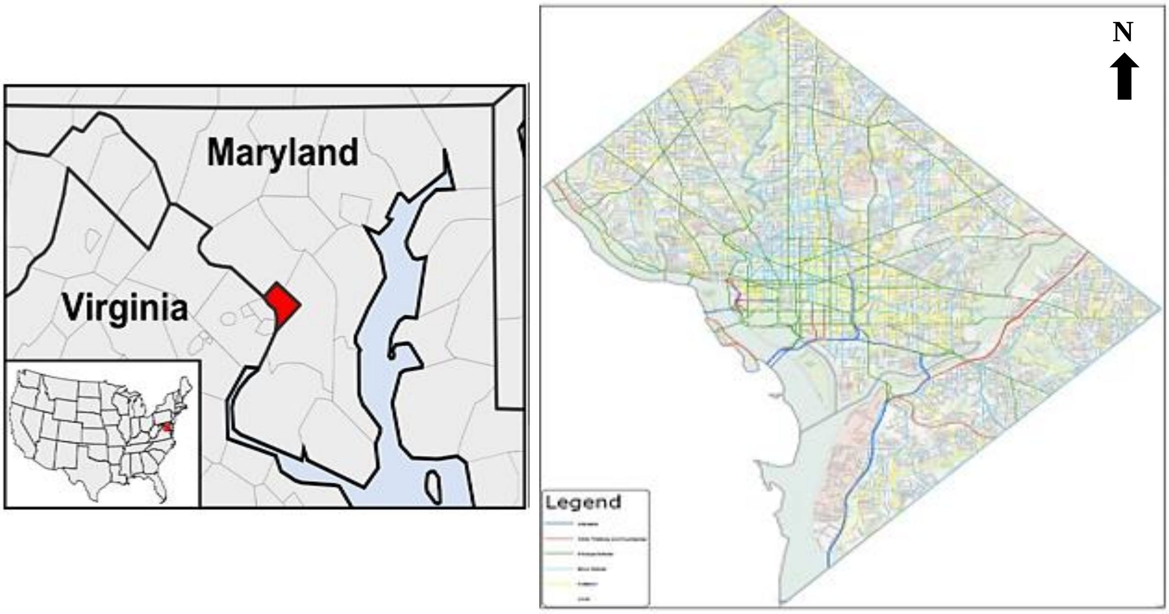


Figure 2. Location and Road map of Washington, DC

WMATA buses have Global Positioning System (GPS) installed onboard of transit buses to track its location and display it on a geographical map of the area (Automatic Vehicle Location). Automatic Passenger Counters (APC) installed on buses count the number of passengers alighting and boarding at each bus stop which helps in the computation of the total number of patrons onboard. For this study, six (6) months (January 2019 – June 2019) worth of AVL and APC data for 6 WMATA bus routes was collected for analysis. Bus routes of two functional roadway classifications were considered: arterials and collectors. In general, bus routes with the following characteristics were considered:

- Routes with high patronage bus stops: data for bus routes with relatively higher patronage was provided by consulted WMATA Officials.
- Routes with bus stops with longer headways: Routes can have several bus stops that accumulate larger groups of patrons boarding or alighting buses. Such stops can account for higher dwell time of buses along the route.
- Routes with bus stops near metro rail stations: Bus stops near metro rail stations usually have a high number of patrons since they provide patrons access to bus services and vice versa.

Based on the criteria, the following bus routes were selected for the study:

1. Route 70 (Georgia Avenue Line)
2. Route 32 (Pennsylvania Avenue Line)
3. Route 52 (14th Street Line)
4. Route 42 (Mount Pleasant Line)
5. Route D4 (Ivy City-Franklin Square Line)
6. Route S1 (16th Street-Potomac Park Line)

Figure 3 represents the operation paths of the selected routes.

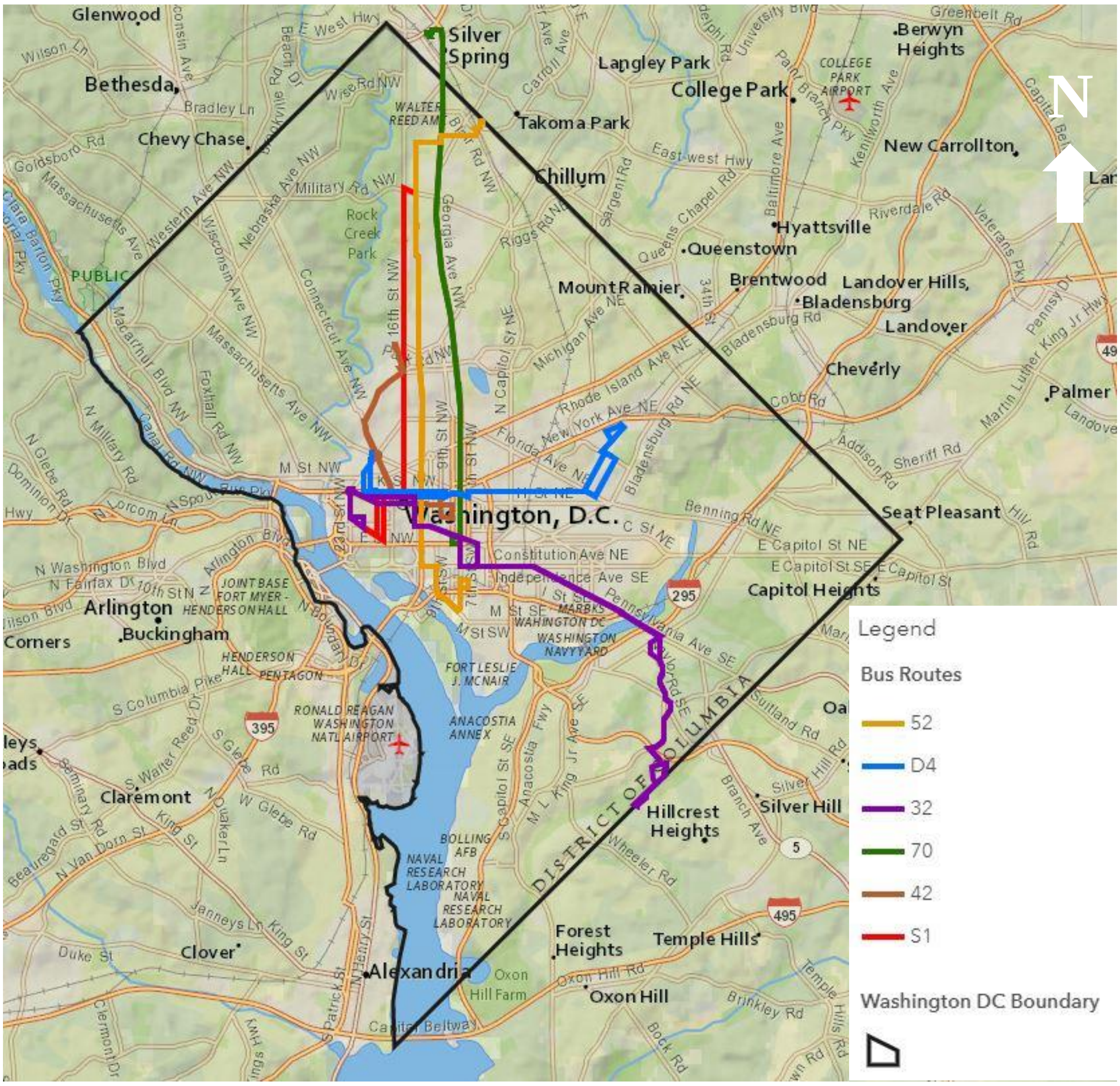


Figure 3. Map of operation paths of selected routes

2.2. Data Extraction

The ANN models were developed based on AVL (Automatic Vehicle Location) and APC (Automatic Passenger Counters) data. Six (6) months (January 2019 – June 2019) worth of AVL and APC data for buses operating in the DC area were used for this research. Excel sheets obtained from WMATA database containing the data from the first week of every month were filtered to obtain only the necessary information required for the analysis. Based on the significance of their impact on travel time of transit buses (from the previous literature), the following independent variables for a bus trip were extracted for each week and for the selected routes from the data:

- Departure and Arrival Times (Event Time)
- Length of Routes
- Passengers Loading/ Unloading
- Dwell Time
- Travel Time
- Location of the bus

The information obtained was used in the Neural Network training. The descriptions of the variables included in the ANN Model are as follows:

- Number of Served Bus Stops Between Origin and Destination (X_1)

A bus can "serve" a stop if there are passengers entering or exiting at that stop. Hence, for the purpose of this research, X_1 is denoted as the number of bus stops between any two "origin" and "destination" points along a bus route. Hence, if a bus serves Stop 1 and Stop 2, $X_1 = 0$. If a bus serves Stop 1 and Stop 5, $X_1 = 3$.

- Length of Route between Bus Stops (X_2)

The data provided by WMATA had the odometer readings of all the buses along a route. Hence, X_2 was obtained by taking the difference of the odometer readings between any two served bus stops.

- Average Number of Passengers in the Bus (X_3)

Automatic Passenger Counting (APC) systems installed on all the WMATA buses count the number of passengers getting in and out of the bus from the front and the rear doors. At any point, the number of passengers will be the sum of passengers already on the bus (excluding the origin bus stop) and the number of patrons boarding the bus minus the number of patrons alighting the bus at any stop. The average number (X_3) can be found by taking the mean of the number of passengers on the bus between any two particular served bus stops.

- Average Dwell Time of served Bus Stops (X_4)

The dwell time is the period during which the front and the rear doors of a bus at a bus stop remained open to serve patrons. Average dwell time (X_4) for the analysis is the mean of dwell times of the served bus stops between and origin and destination point.

- Number of Intersections between Bus Stops (X_5)

Field data observations were conducted to determine the number of intersections (un-signalized and signalized) between any two served bus stops along a route (X_5).

Since traffic characteristics depend on the time of the day, separate ANNs models were developed for AM Peak (7:00 AM - 9:30 AM), PM Peak (4:00 PM - 6:30 PM) and Mid-Day Peak (10:00 AM – 2:30 PM) periods. The general form of the matrix containing the different independent variables and the dependent variable (travel time) used for the neural network analysis is presented in Table 1. For this study, a minimum sample size of 500 origin to destination trips of multiple transit buses on a route were extracted for each peak period from the 6-month AVL/APC data obtained from WMATA. Thus, a minimum sample of 1,500 origin to destination trips were extracted and exported into a Comma Separated Values (CSV) file for analysis of the three peak periods for each bus route.

Table 1. Sample peak period ANN data matrix model for a bus route

Trip ID	No. of SERVED Bus Stops (SBS), X_1	Length of Route (L), X_2	Average Number of Passengers (P), X_3	Average Dwell Time (DT), X_4	No. of Intersections (IT), X_5	Travel Time (TT), Y
1	A	D	G	J	M	P
2	B	E	H	K	N	Q
3..	C	F	I	L	O	R
…	-	-	-	-	-	-
500	S	T	U	V	W	X

2.3. Data Analysis

Neural Designer software was used for the neural network analysis. The software that incorporates data science and machine learning techniques which helps to build, train and deploy neural network models. Analysis for the project involved the following steps.

2.3.1. Descriptive Statistics

Descriptive statistics including the mean, median and standard deviation were computed for the bus travel times as well as the other predictor or independent variables. The averages of predictors such as dwell times and number of passengers per peak periods were also obtained.

2.3.2. ANN Model Development

The purpose of developing an ANN model in this research is to determine the travel time of the bus on a route using approximation technique in Neural Designer Software. In the approximation technique, the neural network learns from the input-target examples provided by the user. It should be noted that the objective of approximation is to produce a neural network which performs well in generalization and makes good prediction for unseen data (good fit) rather than capturing specific details in the data set (overfitting). The software was used to split the data into a training set (75%) and a testing set (25%). The training dataset was used to train and develop the model while the testing dataset was used to validate the model.

Training was conducted through an iterative process of feed forward and errorback propagations until the gradient normalization goal or the stopping criterion of a 1,000 epochs (iterations) was met. The following adjustments were made prior to performing the neural network analysis.

Perceptron Layers

The training of the model was done using a Multilayer Perceptron (MLP). Perceptron layers are important layers that enable the neural network to learn. Numerical values are inputs ($X_1,\ldots,X_n$) for the perceptron neurons in a network to produce a numerical output y (travel time). The result of the output is also affected by the combination of bias (b) and the sum of individual weights of independent variables ($w_1,\ldots,w_n$).

The MLP used for this research consisted of three layers: input layer, hidden layer, and output layer. A typical ANN architecture is presented in Figure 4.

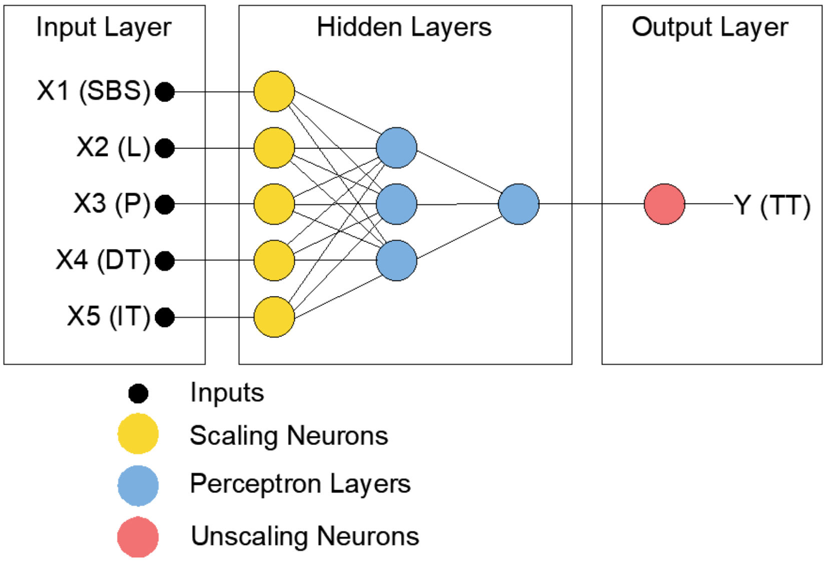


Figure 4. ANN architecture

Data Standardization

The inputs in the data sets did not have the same ranges. Hence, an automatic scaling layer was applied to make the values of all the independent variables comparable. The scaled outputs were unscaled back to the original units using the un-scaling layer on the perceptron layers.

Training Strategy

The training strategy refers to the procedure to carry out the learning process which is applied to neural network. It is done to obtain the minimum possible error in the loss index. The loss index evaluates the performance of a neural network by assessing its parameters. Minimizing the error can be done by finding a set of parameters that fit the neural network to the data set. At the lowest value of the loss index, the gradient is zero. The optimization algorithm enables the capability of varying the parameters to obtain the ideal value for each training iteration or epochs, which gradually decreases the loss. The optimization algorithm stops during training after specific conditions or criteria have been met. Quasi-Newton optimization algorithm was used for training the data sets of all peak periods of the 6 bus routes. The algorithm yields a function with low loss and higher accuracy. The algorithm is the default optimization method in Neural Designer and is also recommended for training medium sized data sets (10-1,000 variables, 1,000-1,000,000 instances).

Model selection in the Neural Designer software program refers to finding the optimal network architecture with the best generalization properties. Order selection was performed to achieve the best selection model that generated the adequate fit to the data provided. The incremental order selection process was used to obtain the optimal order, training and selection errors. Following the order selection, the models were trained to reduce the errors.

2.3.3. Mathematical Equations, Model Testing and Evaluation

Neural Designer was also used to obtain the mathematical equations for approximation of travel times for different peak periods. The errors obtained from conducting the neural network analyses were also documented.

After training the network for the required number of epochs, the models are tested using the set-aside test dataset. Normalized Squared Error (NSE), which is the default error term used when solving approximation problems, was used to evaluate the models. The prediction of data can yield a value between 0 (perfect prediction) and 1 (predicting on basis of the mean). It can be represented by using the following:

$$\mathrm{NSE} = \frac{\sum(\mathrm{O}-\mathrm{T})^2}{NC} \tag{1}$$

Where;

NSE = Normalized Squared Error

O = Outputs

T = Targets

NC = Normalized Coefficient

The Mean Percentage Error (MPE) was also obtained for all the models for each peak period to determine the accuracy. MPE represents the error made in the predicted values as compared to the observed values. It can be represented as

$$\mathrm{MPE} = \frac{1}{n}\frac{\sum|\mathrm{A}-\mathrm{P}|}{A} * 100 \tag{2}$$

Where;

A = Observed values

P = Predicted values

n = Number of observations

3. Results

This section provides the findings of the research.

3.1. Summary Statistics

The distribution of data points that were used to develop the models is presented in Figure 5. An overview of the data of 6 study routes for the AM, PM and Mid-Day Peak hours has been represented.

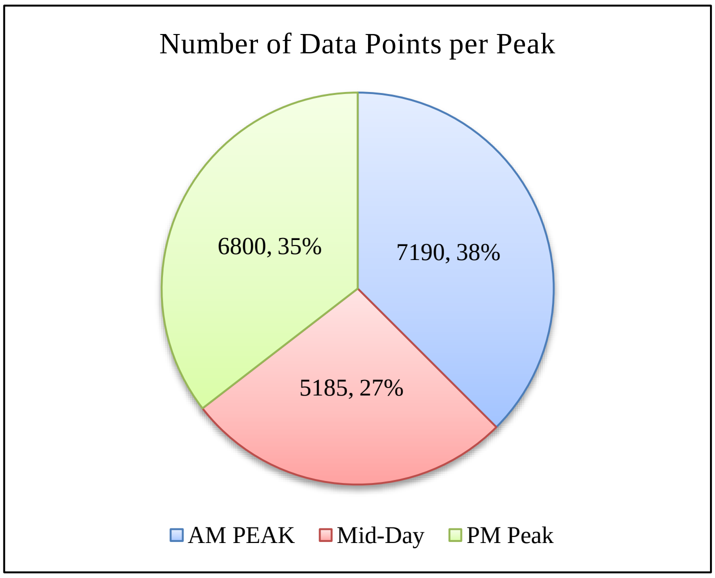


Figure 5. Total number of collected data points per peak period

From Figure 5, a total of 7,190, 5,185 and 6,800 data points were used to develop the neural network models respectively for the AM, Mid-Day and PM Peak periods.

A summary of the average travel time of the buses based on the number of bus stops served during each peak period is presented in Table 2. The descriptive statistics of data sets are presented in Table 3.

Table 2. Average travel time by number of served bus stops

	Average Travel Time (in seconds)		
Number of Served Bus Stops	**AM Peak Period**	**Mid-Day Peak Period**	**PM Peak Period**
0	87.74	104.64	101.54
1	189.21	221.32	218.95
2	287.27	334.10	333.84
3	389.17	447.58	451.05
4	489.92	562.17	571.99
5	590.48	676.73	686.69
6	690.10	791.78	802.48
7	791.66	907.42	919.30
8	885.84	1023.47	1037.80
9	988.84	1140.49	1160.04

Table 3. Descriptive Statistics

	Mean			Standard Deviation		
	AM Peak	**Mid-Day Peak**	**PM Peak**	**AM Peak**	**Mid-Day Peak**	**PM Peak**
SBS (X_1)	3.92	3.94	3.94	2.70	2.71	2.71
L (X_2)	6718.90	6734.20	6016.89	5817.77	5663.62	5186.32
P (X_3)	15.58	12.96	16.92	9.38	6.57	9.71
DT (X_4)	13.81	16.08	17.54	9.92	12.10	17.71
IT (X_5)	15.72	16.71	15.67	10.26	10.88	10.39
TT (Y)	481.64	556.68	562.78	324.42	486.29	401.39

It can be observed from Table 2 that the travel times generally increased over time of the day for all the number of bus stops served (0-9) along the route. The mean and standard deviation of all the variables (dependent and independent) that were used in all three peak periods are presented in Table 3.

3.2. Neural Network Training

This section presents results of the neural network trainings on models to predict the bus Travel Times using Neural Designer software. Matrices for all three peaks were analyzed using the Quasi-Newton algorithms with 2, 3 and 5 number of perceptron layers, separately. The inputs used for all Quasi-Newton Analysis were scaled using the automatic scaling method. The size of the scaling layer was 5 (number of inputs). The scaled outputs were unscaled back to the original units using the unscaling layers on the perceptron layers for all peak periods. Unscaling method for the output layer was performed by using the minimum and maximum method.

Order selection was performed to achieve the best selection model that generated the adequate fit for all the peak periods. The details of the results obtained from neural network analysis using Quasi-Newton algorithm are discussed in the following subsections.

3.2.1. Two-Perceptron Layers

Neural networks were developed for all three peak periods using 2 perceptron layers and analyzed using the Quasi-Newton optimization algorithm. The activation functions of the first layer were set to a hyperbolic tangent while those of second layer were set as linear. Table 4 shows the results of training for all peak periods.

Table 4. Quasi-Newton method results of all peak periods for two perceptron layers (after order selection)

	AM Peak Value	Mid-Day Peak Value	PM Peak Value
Initial Training Error	7.670	60.580	48.390
Final Training Error	0.138	0.052	0.218
Initial Selection Error	8.250	46.510	51.460
Final Selection Error	0.145	0.041	0.230
Epochs Number	615	336	308
Elapsed Time (min:sec)	00:03	00:01	00:01
Stopping Criterion	Gradient Norm Goal	Gradient Norm Goal	Gradient Norm Goal

It can be observed from Table 4 that for the AM Peak period, the initial training error value of 7.67 decreased to 0.138 after 615 epochs. The initial value of the selection error for the AM Peak period was 8.25, and the final value after 615 epochs decreased to 0.145. For the Mid-Day Peak period, the initial training error of 60.58 decreased to 0.0518 after 336 epochs. Similarly, after 336 epochs, the initial value of selection error for the Mid-Day Peak period decreased from 46.51 to 0.0405. For the PM Peak period, initial training error decreased from 48.39 to 0.218 after 308 epochs and the initial selection error decreased from 51.46 to 0.2295.

The highest final training and selection errors were observed for the PM Peak model while the Mid-Day Peak model had the lowest final training and selection errors. The selection error for the Mid-Day Peak period had the greatest change in error (99.913%) while the lowest change in error (98.243%) was obtained for the selection error of the AM Peak model after order selection.

3.2.2. Three-Perceptron Layers

Neural networks for all three peak periods having 3 perceptron layers were also modeled and analyzed using the Quasi-Newton Optimization Algorithm. The activation functions of the first two layers were set to hyperbolic tangent while those of third layer of each peak period were set to linear. Table 5 shows the results of training for all peak periods.

Table 5. Quasi-Newton method results of all peak periods for three perceptron layers (after order selection)

	AM Peak Value	Mid-Day Peak Value	PM Peak Value
Initial Training Error	16.170	93.280	18.000
Final Training Error	0.145	0.053	0.205
Initial Selection Error	16.080	72.120	18.320
Final Selection Error	0.154	0.042	0.229
Epochs Number	812	404	1,000
Elapsed Time (min:sec)	00:04	00:01	00:06
Stopping Criterion	Gradient Norm Goal	Gradient Norm Goal	Maximum number of iterations

It can be observed from Table 5 that for the AM Peak period, the initial training error value of 16.17 decreased to 0.145 after 821 epochs. The initial value of the selection error for the AM Peak period was 16.08, and the final value after 821 epochs decreased to 0.154. From Table 5 it can also be observed that for the Mid-Day Peak period, the initial training error of 93.28 decreased to 0.0532 after 404 epochs. Similarly, after 404 epochs, the initial value of selection error for the Mid-Day Peak period decreased from 72.12 to 0.0415. For the PM Peak period, initial training error decreased from 18.00 to 0.205 after 1,000 epochs and the initial selection error decreased from 18.32 to 0.229.

The highest final training and selection errors were observed for the PM Peak period while the Mid-Day Peak period had the lowest final training and selection errors. The selection error for the Mid-Day Peak had the greatest change in error (99.942%) while the lowest change in error (98.75%) was obtained for the selection error of the PM Peak period after order selection.

3.2.3. Five-Perceptron Layers

Neural networks for all three peak periods having 5 perceptron layers were also modeled and analyzed using the Quasi-Newton Optimization Algorithm. The activation functions of the first four layers were set to hyperbolic tangent while those of last layer of each peak period were set to linear. Table 6 shows the results of training for all peak periods

Table 6. Quasi-Newton method results of all peak periods for five perceptron layers (after order selection)

	AM Peak Value	Mid-Day Peak Value	PM Peak Value
Initial Training Error	96.800	59.490	61.080
Final Training Error	0.150	0.094	0.235
Initial Selection Error	96.640	56.440	65.750
Final Selection Error	0.159	0.106	0.257
Epochs Number	1,000	1,000	927
Elapsed Time (min:sec)	00:05	00:03	00:05
Stopping Criterion	Maximum number of iterations	Maximum number of iterations	Gradient Norm Goal

It can be observed from Table 6 that for the AM peak period, the initial training error value of 96.80 decreased to 0.150 after 1,000 epochs. The initial value of the selection error for the AM Peak period was 96.64, and the final value after 1,000 epochs decreased to 0.159. From Table 6, it can also be observed that for the Mid-Day Peak period, the initial training error of 59.49 decreased to 0.094 after 1,000 epochs. Similarly, after 1,000 epochs, the initial value of selection error for the Mid-Day Peak period decreased from 56.44 to 0.106. For the PM Peak period, initial training error decreased from 61.08 to 0.235 after 1,000 epochs and the initial selection error decreased from 65.75 to 0.257. The highest final training and selection errors were observed in the PM Peak model while the Mid-Day Peak period had the lowest final training and selection errors. The training error for the AM Peak period had the highest change in error (99.845%) while the lowest change in error (99.638%) was obtained for the selection error of the PM Peak period after order selection.

3.3. Errors and Error Statistics

This section presents the comparison of Normalized Squared Errors and the error statistics after performing neural network training.

3.3.1. Normalized Squared Error

The normalized squared errors were obtained for all the instances which evaluated the model for each use. Table 7 presents the normalized squared errors for the training, selection and testing instances that were obtained from the model for all number of perceptron layers.

Table 7. Normalized squared errors for training, selection and testing instances (2 layer quasi-newton method)

Two Perceptron Layers			
	AM Peak	Mid-Day Peak	PM Peak
Training Error	0.138	0.052	0.218
Selection Error	0.145	0.041	0.229
Testing Error	0.147	0.053	0.257
Three Perceptron Layers			
Training Error	0.145	0.053	0.205
Selection Error	0.154	0.042	0.229
Testing Error	0.163	0.055	0.254
Five Perceptron Layers			
Training Error	0.150	0.094	0.212
Selection Error	0.159	0.106	0.225
Testing Error	0.169	0.119	0.258

It can be observed from Table 7 that the lowest testing errors were obtained for the Mid-Day Peak model. Figures 6 through 8 present the NSE obtained for the training, selection and testing errors for AM, Mid-Day and PM Peak periods, respectively.

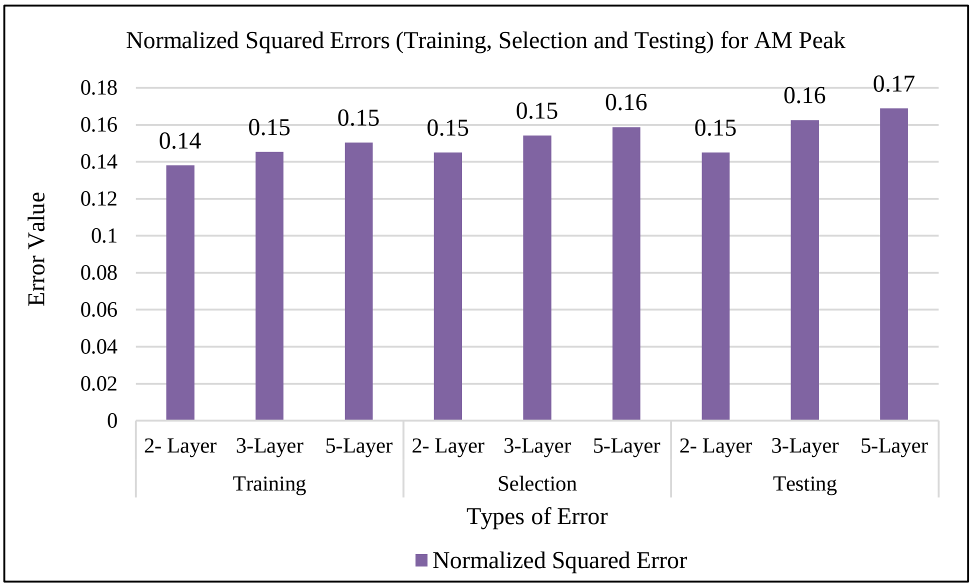


Figure 6. Normalized squared errors obtained for AM peak period neural network analyses

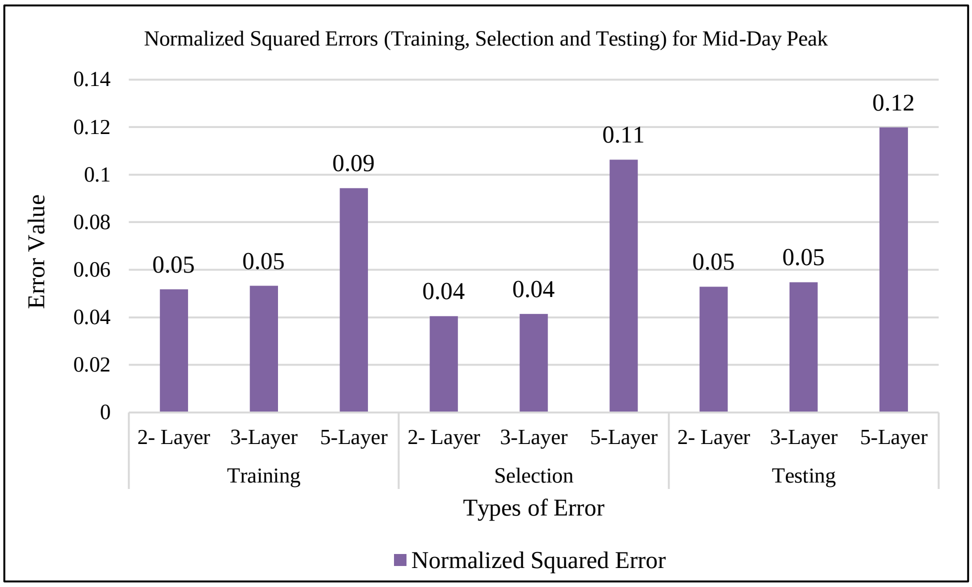


Figure 7. Normalized squared errors obtained for Mid-Day peak period neural network analyses

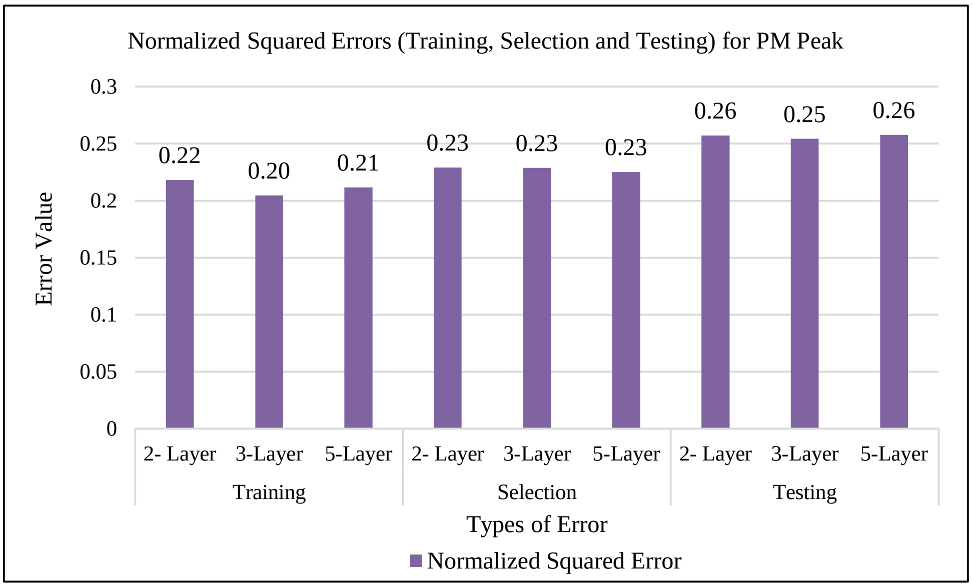


Figure 8. Normalized squared errors obtained for PM peak period neural network analyses

From the graphs, it can be observed that the highest NSE for the training, selection and testing errors of all peak models were generally obtained for the models with 5 perceptron layers. The models with 2 perceptron layers undergoing Quasi-Newton method for the AM and Mid-Day Peak periods produced the lowest NSE for training, selection and testing sets.

3.3.2. Error Statistics

The Mean Absolute Errors (MAE) and Mean Percentage Errors (MPE) were also obtained for the models during the neural network analyses. MAE and MPE are a measure of errors and computed average of percentage errors between paired observations which can be used as a determination of model's accuracy. Figure 9 represents the MAE and MPE for all the 3 peak period models.

The Mean Absolute Error parameter measures the difference between two continuous variables. The prediction error, i.e. the difference between the observed and predicted value, is converted to positive to give the absolute error [18]. The mean (average sum) of all recorded absolute errors (MAE), gives an idea of the average error to expect from the prediction model. However, the MAE does not provide the relative size of the error especially when comparing several models. The Mean Percentage Error (MPE) allows for the computation of the mean error in percentage terms. This determines how big or small an error is and provides a better means of comparing the various models [19].

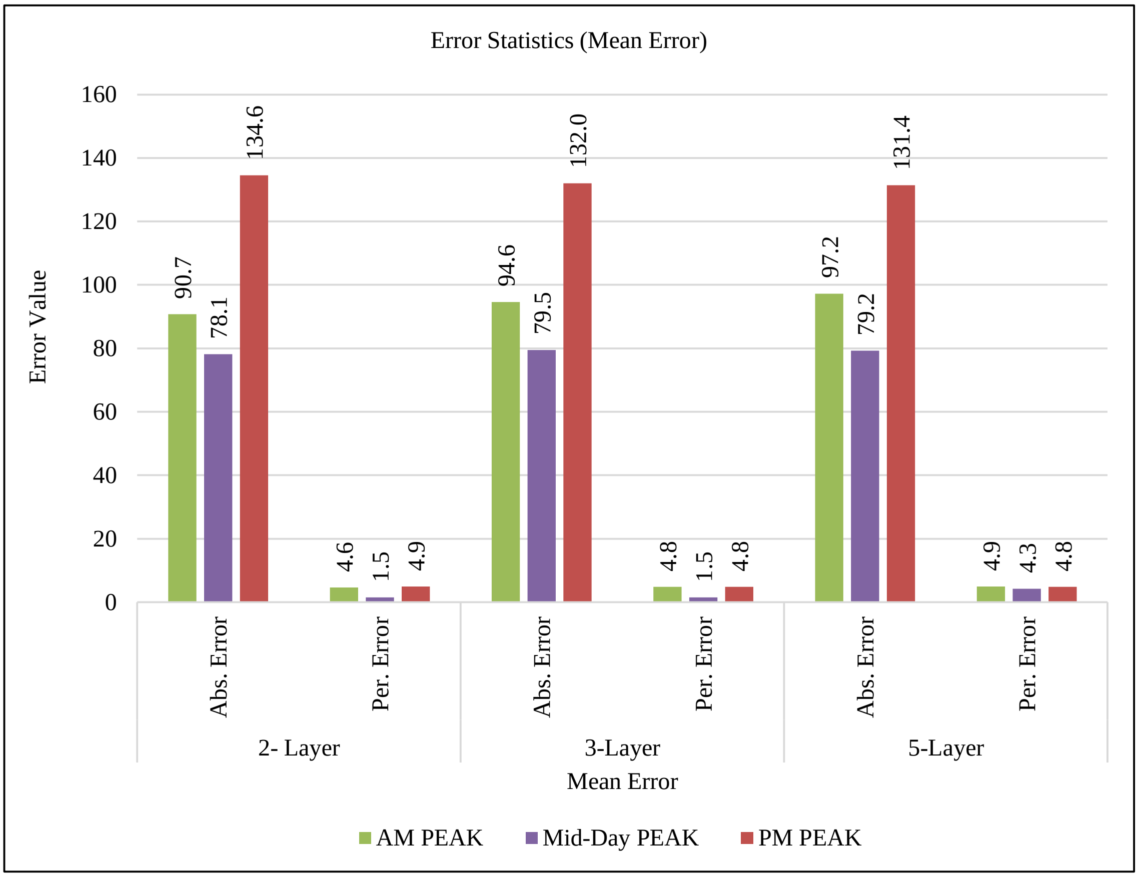


Figure 9. Mean absolute and mean percentage errors obtained after neural network analyses

It can be observed from Figure 9 that the lowest Mean Absolute Errors for the AM and Mid-Day Peak periods were obtained in the Quasi-Newton model with 2 layers whereas the lowest MAE for the PM Peak period was obtained in the Quasi-Newton model with 5 layers. The models with 5 layers displayed the highest overall Mean Percentage Errors

Since lower training error corresponds to overfitting, it may not be a good validation of the predictive model. Overfitting leads to inaccuracy in predicting the correct output from unseen data while testing. Hence, testing error was analyzed for all the models to evaluate the accuracy of approximation. The mean absolute error and the mean percentage error were compared to test the quality of the predictive models. It could be observed that the models trained using Quasi-Newton Algorithm having 2 perceptron layers had the lowest normalized squared errors (testing) followed by training models having 3 perceptron layers. Mid-Day Peak models had the lowest overall errors for both training algorithms.

3.4. Mathematical Expressions

Table 8 presents the mathematical expressions that were obtained from the neural network analysis for all peak models undergoing Quasi-Newton Analyses.

Table 8. Neural network analysis output equations

Travel Time	Equation
AM	0.5*(scaled_Y$_{AM}$+1.0)*(1976)+2
Mid-Day	0.5*(scaled_Y$_{Mid\text{-}Day}$+1.0)*(5174)+6
PM	(scaled_Y$_{PM}$+1.0)*(2739)+21

Table 8 represents general equations for all three peak periods for the different analyses (with multiple perceptron layers). It should be noted that the scaled values of Y$_{AM}$, Y$_{Mid\text{-}Day}$ and Y$_{PM}$ change with different analysis. The mathematical expressions for the scaled Y outputs that were obtained from the neural network analysis for all peak models undergoing Quasi-Newton Analysis can be represented as:

- ***Two Perceptron Layers***

scaled_Y$_{AM}$ = (-0.403933+ (tanh (0.266595+ (sA_ X$_1$*0.159083)+ (sA_ X$_2$*1.38412) + (sA_ X$_3$*0.773302)+ (sA_ X$_4$*-0.10047)+ (sA_ X$_5$*-1.5879))*-0.827554)+ (tanh (0.763357+ (sA_X$_1$*0.282538)+ (sA_X$_2$*-0.321875)+ (sA_X$_3$*-0.425407)+ (sA_X$_4$*-0.076593)+ (sA_X$_5$*-0.562577))*0.475845)+ (tanh (-0.773348+ (sA_X$_1$*-0.104466)+ (sA_X$_2$*-0.524453)+ (sA_X$_3$*0.331032)+ (sA_X$_4$*0.0637511)+ (sA_X$_5$*0.78019))*1.51826)+ (tanh (-0.731044+ (sA_X$_1$*-1.3292)+ (sA_X$_2$*0.174948)+ (sA_X$_3$*-0.603121)+ (sA_X$_4$*-0.132371)+ (sA_X$_5$*1.4993))*0.184144)+ (tanh (-0.702817+ (sA_X$_1$*-0.8047)+ (sA_X$_2$*1.77398) + (sA_X$_3$*-0.152839)+ (sA_X$_4$*0.245255)+ (sA_X$_5$*-0.334036))*-0.241722)+(tanh (-0.61739+ (sA_X$_1$*0.152483)+ (sA_X$_2$*-0.725123)+ (sA_X$_3$*0.380162)+ (sA_X$_4$*0.0446854)+ (sA_X$_5$*0.548037))*-1.17179)+ (tanh (-0.26219+ (sA_X$_1$*0.0512189)+ (sA_X$_2$*1.25502)+ (sA_X$_3$*-1.05877)+ (sA_X$_4$*0.0268656)+ (sA_X$_5$*-0.638987))*0.385873)+ (tanh (-0.508739+ (sA_X$_1$*-0.225638)+ (sA_X$_2$*-1.86452)+ (sA_X$_3$*-0.810278)+ (sA_X$_4$*0.075467)+ (sA_X$_5$*1.91736))*-0.688366)+ (tanh (-0.0139338+ (sA_X$_1$*-0.0619438)+(sA_X$_2$*-0.703159)+(sA_X$_3$*0.699206)+(sA_X$_4$*-0.0176063)+ sA_X$_5$*0.224042))*0.695668)+ (tanh (-0.0315544+ (sA_X$_1$*0.132696)+ (sA_X$_2$*0.19984)+ (sA_X$_3$*0.0444743)+ (sA_X$_4$*0.0152423)+ (sA_X$_5$*-0.129342))*2.37657))

scaled_Y$_{Mid\text{-}Day}$ = (-0.208024+ (= tanh (-0.0899416+ (sM_X$_1$*-0.902712)+ (sM_X$_2$*0.773495)+ (sM_X$_3$*-0.213636)+ (sM_X$_4$*-0.0126037)+(sM_X$_5$*0.312714))*-0.159253)+ (tanh (1.09159+ (sM_X$_1$*0.335772)+ (sM_X$_2$*-0.346538)+ (sM_X$_3$*0.052857)+ (sM_X$_4$*-0.0197632)+ (sM_X$_5$*-0.00913458))*-1.66349)+ (tanh (1.68263+ (sM_X$_1$*0.646806)+ (sM_X$_2$*-0.783611)+ (sM_X$_3$*-0.00646778)+ (sM_X$_4$*-0.0243315)+ (sM_X$_5$*0.0401995))*0.801582) + (tanh (-0.0188228+ (sM_X$_1$*0.051862)+ (sM_X$_2$*0.0738155)+ (sM_X$_3$*0.00713647)+ (sM_X$_4$*-0.0134423)+ (sM_X$_5$*0.032604))*1.1423));

scaled_Y$_{PM}$ = (-0.109188+ (tanh (-0.125552+ (sP_X$_1$*-0.330932)+ (sP_X$_2$*1.98228)+ (sP_X$_3$*-0.210823)+ (sP_X$_4$*0.12685)+(sP_X$_5$*-1.17731))*1.08879)+ (tanh (0.42971+ (sP_X$_1$*-0.113247)+ (sP_X$_2$*1.63058)+ (sP_X$_3$*-0.0563956)+ (sP_X$_4$*0.0776343)+ (sP_X$_5$*-1.27139))*-2.69854)+ (tanh (0.892756+ (sP_X$_1$*-0.13717)+ (sP_X$_2$*0.114125)+ (sP_X$_3$*-0.114468)+ (sP_X$_4$*-0.0215025)+ (sP_X$_5$*-0.472637))*-0.551765)+ (tanh (0.339223+ (sP_X$_1$*-0.0539536)+ (sP_X$_2$*-0.784304)+ (sP_X$_3$*-0.0873344)+ (sP_X$_4$*-0.0322945)+ (sP_X$_5$*0.837868))*-2.13805)+ (tanh (-1.13634+ (sP_X$_1$*0.161904)+ (sP_X$_2$*-2.72193)+ (sP_X$_3$*0.244611)+ (sP_X$_4$*-0.113411)+ (sP_X$_5$*1.99965))*-0.937324)+ (tanh (1.35152+ (sP_X$_1$*0.0724538)+ (sP_X$_2$*-1.50671)+ (sP_X$_3$*-0.0274335)+ (sP_X$_4$*-0.0893173)+ (sP_X$_5$*1.62717))*1.2105));

The scaled values of independent variables are provided in Table 9.

Table 9. Scaled values of independent variables

AM Peak	Mid-Day Peak	PM Peak
sA_X$_1$ = (X$_1$-3.92349)/2.70379;	sM_X$_1$ = (X$_1$-3.94118)/2.71125;	sP_X$_1$ = (X$_1$-3.94118)/2.71119;
sA_X$_2$ = (X$_2$-6718.9)/5817.77;	sM_X$_2$ = (X$_2$-6734.2)/5663.62;	sP_X$_2$ = (X$_2$-6016.89)/5186.32;
sA_X$_3$ = (X$_3$-15.5805)/9.37685;	sM_X$_3$ = (X$_3$-12.9582)/6.56694;	sP_X$_3$ = (X$_3$-16.9162)/9.71334;
sA_X$_4$ = (X$_4$-13.8079)/9.92248;	sM_X$_4$ = (X$_4$-16.0783)/12.0996;	sP_X$_4$ = (X$_4$-17.5412)/17.7121;
sA_X$_5$ = (X$_5$-15.7173)/10.2624;	sM_X$_5$ = (X$_5$-16.7128)/10.8819;	sP_X$_5$ = (X$_5$-15.6674)/10.3884;

- ***Three Perceptron Layers***

scaled_Y_{AM} = (0.215174+ (tanh (-0.536023+ (y_1_1*1.51243)+ (y_1_2*0.759729)+ (y_1_3*-1.32225))*-0.975811)+ (tanh (-1.94384+ (y_1_1*-0.396438)+ (y_1_2*-1.70619)+ (y_1_3*0.291821))*1.23331)+ (tanh (1.4332+ (y_1_1*0.702253)+ (y_1_2*0.251199)+ (y_1_3*1.03667))*1.12438)+ (tanh (0.0512586+ (y_1_1*-0.767183)+ (y_1_2*-0.512971)+ (y_1_3*-0.253243))*-1.60024)+ (tanh (0.756908+ (y_1_1*-0.150335)+ (y_1_2*0.819108)+ (y_1_3*1.99338))*-1.04541)+ (tanh(1.54649+ (y_1_1*0.116734)+ (y_1_2*-1.85288)+ (y_1_3*-1.35402))*1.33817)+ (tanh (0.0243083+ (y_1_1*0.374252)+ (y_1_2*0.328831)+ (y_1_3*0.181334))*0.271241)+ (tanh (-0.125053+ (y_1_1*-0.328579)+ (y_1_2*-0.256182)+ (y_1_3*0.590709))*-0.351073)+ (tanh (-1.88825+ (y_1_1*-1.69526)+ (y_1_2*0.331318)+ (y_1_3*1.16758))*1.20335));

scaled_$Y_{Mid\text{-}Day}$=(0.103787+ tanh (-0.536023+ (y_1_1*1.51243)+ (y_1_2*0.759729)+ (y_1_3*-1.32225))*1.6249));

scaled_Y_{PM} = (0.684445+ (tanh (0.45443+ (y_1_1*2.68585)+ (y_1_2*0.0347495)+ (y_1_3*-0.817478))*1.35074)+ (tanh (-0.108927+ (y_1_1*-0.713266)+ (y_1_2*0.417959)+ (y_1_3*1.17019))*-1.28872)+ (tanh (-0.322299+ (y_1_1*-1.97763)+ (y_1_2*0.308789)+ (y_1_3*1.17455))*2.40162)+ (tanh (-0.587322+ (y_1_1*-1.8385)+ (y_1_2*-0.587654)+ (y_1_3*-1.136))*1.26585)+ (tanh (2.47199+ (y_1_1*-0.100312)+ (y_1_2*0.0381556)+ (y_1_3*-2.21539))*-1.08219)+ (tanh (0.550239+ (y_1_1*1.09962)+ (y_1_2*0.698722)+ (y_1_3*1.51868))*1.46912)+ (tanh (1.43996+ (y_1_1*-0.139531)+ (y_1_2*-1.19197)+ (y_1_3*-1.27155))*-0.888777)+ (tanh (-2.5035+ (y_1_1*-2.03767)+ (y_1_2*1.91548)+ (y_1_3*2.29532))*-1.32063)+ (tanh (-0.283164+ (y_1_1*-1.63158)+ (y_1_2*1.42752)+ (y_1_3*1.34363))*-0.494677)+(tanh (-2.36089+ (y_1_1*-2.65308)+ (y_1_2*1.56991)+ (y_1_3*2.56494))*1.01097));

The scaled values of y_x_x for all peak periods are provided in Table 10.

Table 10. Scaled y values for 3 perceptron layer analysis

	Mid-Day Peak	**PM Peak**
y_1_1 = tanh (-0.528509+ (sA_X_1*0.326117)+ (sA_X_2*0.272163)+ (sA_X_3*-0.161073)+ (sA_X_4*-0.0182433)+ (sA_X_5*-0.332747))	y_1_1 = tanh (1.26074+ (sM_X_1*0.340538)+ (sM_X_2*-0.388042)+ (sM_X_3*0.0323315)+ (sM_X_4*-0.0204073)+ (sM_X_5*0.0650814))	y_1_1 = tanh (-0.251511+ (sP_X_1*0.257925)+ (sP_X_2*-1.10134)+ (sP_X_3*0.122933)+ (sP_X_4*-0.0966804)+ (sP_X_5*0.802904));
y_1_2 = tanh (0.309257+ (sA_X_1*-0.087491)+ (sAX_2*-0.489761)+ (sA_X_3*-0.0127571)+ (sA_X_4*-0.0235167)+ (sA_X_5*0.0684882))	y_1_2 = tanh (0.31964+ (sM_X_1*0.13657)+ (sM_X_2*0.0305903) + (sM_X_3*0.0239318)+ (sM_X_4*-0.0102187)+ (sM_X_5*0.020976))	y_1_2 = tanh (0.149037+ (sP_X_1*-1.18999)+ (sP_X_2*-0.0809956)+ (sP_X_3*-0.885189)+ (sP_X_4*0.0770171)+ (sP_X_5*0.677185))
y_1_3 = tanh (-0.292626+ (sA_X_1*0.234084)+ (sA_X_2*-1.68782)+ (sA_X_3*0.00988481)+ (sA_X_4*0.0292022)+ (sA_X_5*1.51933))	y_1_3 = tanh (-1.69767+ (sM_X_1*-0.570229)+ (sM_X_2*0.803204)+ (sM_X_3*0.0739754)+ (sM_X_4*0.0383746)+ (sM_X_5*-0.180737))	y_1_3 = tanh (0.0309167+ (sP_X_1*0.278205)+ (sP_X_2*-0.199602)+ (sP_X_3*0.142916)+ (sP_X_4*-0.0180635)+ (sP_X_5*0.414978))

The scaled values of X_1 to X_5 used in the calculations for Quasi-Newton methods with 3 perceptron layers were same as the scaled values used for all peak period calculations with 2 perceptron layers.

- ***Five Perceptron Layers***

scaled_Y_{AM} = (-0.154104+ (tanh (-0.0548965+ (tanh (0.0129885+ (y_2_1*0.0948825)+ (y_2_2*0.484642)+ (y_2_3*0.449509))*0.904467)+(tanh(0.158871+(y_2_1*-1.1456)+(y_2_2*0.627338)+ (y_2_3*0.588445))*1.32304)+ (tanh (0.12641+ (y_2_1*-0.0131531)+ (y_2_2*0.13634)+ (y_2_3*0.320951))*0.295127))*-1.67074));

scaled_$Y_{Mid\text{-}Day}$ = (0.217368+ (tanh (0.117475+ (tanh (0.406982+ (y_2_1*-1.36989)+ (y_2_2*0.220252)+ (y_2_3*0.415707))*1.27176)+ (tanh (0.206687+ (y_2_1*-1.42396)+ (y_2_2*-0.144306)+ (y_2_3*0.357474))*-1.66689)+ (tanh (0.120252+ (y_2_1*0.87812)+ (y_2_2*-0.698861)+ (y_2_3*1.36139))*-1.6817))*2.53726));

scaled_Y_{PM} = (0.870799+ (tanh (-0.0324586+ (tanh (-1.18555+ (y_2_1*-3.3865)+ (y_2_2*2.00885)+ (y_2_3*-0.6039))*0.642708)+ (tanh (-1.0163+ (y_2_1*-1.22273)+ (y_2_2*0.853594)+ (y_2_3*0.971635))*0.479974)+ (tanh (-1.02954+ (y_2_1*-0.0673815)+ (y_2_2*0.376433)+ (y_2_3*-1.03609))*-0.0206083))*-0.468192)+ (tanh (0.34681+ (tanh (-1.18555+ (y_2_1*-3.3865)+ (y_2_2*2.00885)+ (y_2_3*-0.6039))*0.587993)+ (tanh (-1.0163+ (y_2_1*-1.22273)+(y_2_2*0.853594)+(y_2_3*0.971635))*-0.853453)+(tanh(-1.02954+(y_2_1*-0.0673815)+ (y_2_2*0.376433) + (y_2_3*-1.03609))*2.01037))*2.5343));

The scaled values of y_x_x for all peak periods are provided in Table 11.

Table 11. Scaled y values for 5 perceptron layer analysis

Values for y_2_x		
AM Peak	**Mid-Day Peak**	**PM Peak**
y_2_1 = tanh (0.277517+ (y_1_1*1.31471)+ (y_1_2*-0.693054)+ (y_1_3*-0.345595));	y_2_1 = tanh (0.602469+ (y_1_1*1.60689)+ (y_1_2*1.75895)+ (y_1_3*-0.870177));	y_2_1 = tanh (-0.381238+ (y_1_1*-0.989311)+ (y_1_2*-0.921348)+ (y_1_3*2.04451));
y_2_2 = tanh (-0.548299+ (y_1_1*1.04299)+ (y_1_2*1.14654)+ (y_1_3*-0.425578));	y_2_2 = tanh (0.200869+ (y_1_1*1.02723)+ (y_1_2*0.0400969)+ (y_1_3*-0.196134));	y_2_2 = tanh (-0.091402+ (y_1_1*-0.388481)+ (y_1_2*-0.993532)+ (y_1_3*3.53479));
y_2_3 = tanh (0.293222+ (y_1_1*0.0230052)+ (y_1_2*0.651744)+ (y_1_3*0.0107091));	y_2_3 = tanh (0.420943+ (y_1_1*0.291185)+ (y_1_2*-0.53263)+ (y_1_3*-0.6235));	y_2_3 = tanh (0.0898576+ (y_1_1*-0.527846)+ (y_1_2*-0.040224)+ (y_1_3*0.128471));
Values for y_1_x		
y_1_1 = tanh (-0.992594+ (sA_X$_1$*0.173972)+ (sA_X$_2$*0.0526949)+ (sA_X$_3$*-0.288492)+ (sA_X$_4$*-0.054571)+ (sA_X$_5$*-0.518312));	y_1_1 = tanh (-0.843909+ (sM_X$_1$*-0.274577)+ (sM_X$_2$*1.00172)+ (sM_X$_3$*0.0866354)+ (sM_X$_4$*-0.0624104)+ (sM_X$_5$*-0.798224));	y_1_1 = tanh (0.126043+ (sP_X$_1$*0.0549122)+ (sP_X$_2$*0. 0487273)+ (sP_X$_3$*0.0244436)+ sP_X$_4$*0.00276169)+ (sP_X$_5$*0.114946));
y_1_2 = tanh (0.0464584+ (sA_X$_1$*-0.00771148)+ (sA_X$_2$*-0.0880088)+ (sA_X$_3$*-0.0390799)+ (sA_X$_4$*-0.0136223)+ (sA_X$_5$*-0.0669074));	y_1_2 = tanh (1.64865+ (scaled_X$_1$*0.382326)+ (sM_X$_2$*-1.53801) + (sM_X$_3$*-0.400741)+ (sM_X$_4$*0.160774)+ (sM_X$_5$*1.66477));	y_1_2 = tanh (3.0596+ (sP_X$_1$*2.37948)+ (sP_X$_2$*-0.753125)+ (sP_X$_3$*1.67366)+ (sP_X$_4$*-0.14746)+ (sP_X$_5$*-2.2107));
y_1_3 = tanh (-1.90315+ (sA_X$_1$*-0.843687)+ (sA_X$_2$*2.99375)+ (sA_X$_3$*0.307466)+ (sA_X$_4$*0.0124248)+ (sA_X$_5$*-2.3659));	y_1_3 = tanh (0.375072+ (sM_X$_1$*0.0209689)+ (sM_X$_2$*0. 197518)+ (sM_X$_3$*0.0124373)+ (sM_X$_4$*-0.0107245)+ (sM_X$_5$*-0.0832514));	y_1_3 = tanh (0.394575+ (sP_X$_1$*-0.046241)+ (sP_X$_2$*0. 791731)+ (sP_X$_3$*-0.0749282)+ (sP_X$_4$*0.0879477)+ (sP_X$_5$*-0.594842));

The scaled values of X_1 to X_5 used in the calculations for Quasi-Newton methods with 5 perceptron layers were same as the scaled values used for all peak period calculations with 2 perceptron layers.

4. Conclusion

Neural Network models were developed in this research which can be potentially helpful for transit agencies to improve bus travel time prediction. Delivering real time information pertaining to travel times of buses conveniently to the patrons will ensure the reliability of such transit services. Hence, improvement in the credibility and performance of the WMATA online NextBus Arrival service can increase ridership amongst patrons and decrease the dependency of private owned vehicles. The short and long-term benefits other than improvement in bus travel times include alleviation of traffic congestion, reduction of travel times for overall road users in urban areas like Washington, DC and decrease of vehicular emission. Moreover, the prediction models can serve as an excellent tool to build schedules for new bus routes in or around Washington, DC. The results of the analyses indicate that ANN models can effectively predict travel times of buses on selected routes with minimal percentage errors. From the results, it can be observed that the highest MPE was observed for the PM Peak model with 2 perceptron layers (4.9%). The value is lower than the lowest MPE value (8.19%) which was obtained in the study conducted by Yin et al. [6]. The ANN models could be incorporated into several other predictive models used by WMATA to provide patrons with travel time information online or at bus stops. These models could be adopted by transit agencies in other jurisdictions with similar characteristics to that of the Washington, DC area.

5. Acknowledgements

The authors extend their appreciation to the U.S. Department of Transportation's University Transportation Centers Program and the staff of the Mineta Transportation Institute for funding and contributing to this study. Any opinions, findings, and conclusions or recommendations in this material are those of the authors and do not necessarily reflect the views of the U.S. Department of Transportation.

6. Conflicts of Interest

The authors declare no conflict of interest.

7. References

[1] Transportation in Washington, D.C., Wikipedia. Available online: https://en.wikipedia.org/w/index.php?title=Transportation_in_Washington,D.C.&oldid=960055612. (Accessed on 31 May 2020).

[2] "Metro System," Mar. 13, 2020. Available online: https://www.commuterpage.com/ways-to-get-around/metro-system/ (Accessed on 13 March 2020).

[3] "Metrobus (Washington, D.C.)," Wikipedia. Available online: https://en.wikipedia.org/w/index.php?title=Metrobus_(Washington,_D.C.)&oldid=959888008. (Accessed on 31 May 2020).

[4] Abdulrazzaq, Layth Riyadh, Mohammed Naeem Abdulkareem, Muhamad Razuhanafi Mat Yazid, Muhamad Nazri Borhan, and Mina Salah Mahdi. "Traffic Congestion: Shift from Private Car to Public Transportation." Civil Engineering Journal 6, no. 8 (August 1, 2020): 1547–1554. doi:10.28991/cej-2020-03091566.

[5] Arhin, Stephen, Errol C. Noel, and Mineta National Transit Research Consortium. Evaluation of bus transit reliability in the District of Columbia. No. CA-MNTRC-13-1139. Mineta Transportation Institute, 2013.

[6] Yin, Tingting, Gang Zhong, Jian Zhang, Shanglu He, and Bin Ran. "A Prediction Model of Bus Arrival Time at Stops with Multi-Routes." Transportation Research Procedia 25 (2017): 4623–4636. doi:10.1016/j.trpro.2017.05.381.

[7] Ranjitkar, Prakash, Li-Sian Tey, Enakshi Chakravorty, and Kirsten L. Hurley. "Bus Arrival Time Modeling Based on Auckland Data." Transportation Research Record: Journal of the Transportation Research Board 2673, no. 6 (April 16, 2019): 1–9. doi:10.1177/0361198119840620.

[8] Jeong, R., and R. Rilett. "Bus Arrival Time Prediction Using Artificial Neural Network Model." Proceedings. The 7th International IEEE Conference on Intelligent Transportation Systems (IEEE Cat. No.04TH8749) (2004). doi:10.1109/itsc.2004.1399041.

[9] Chien, Steven I-Jy, Yuqing Ding, and Chienhung Wei. "Dynamic bus arrival time prediction with artificial neural networks." Journal of transportation engineering 128, no. 5 (2002): 429-438. doi: 10.1061/(ASCE)0733-947X(2002)128:5(429).

[10] Yu, Zhengyao, Jonathan S. Wood, and Vikash V. Gayah. "Using Survival Models to Estimate Bus Travel Times and Associated Uncertainties." Transportation Research Part C: Emerging Technologies 74 (January 2017): 366–382. doi:10.1016/j.trc.2016.11.013..

[11] Treethidtaphat, Wichai, Wasan Pattara-Atikom, and Sippakorn Khaimook. "Bus Arrival Time Prediction at Any Distance of Bus Route Using Deep Neural Network Model." 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC) (October 2017). doi:10.1109/itsc.2017.8317891.

[12] Chen, Chi-Hua. "An Arrival Time Prediction Method for Bus System." IEEE Internet of Things Journal 5, no. 5 (October 2018): 4231–4232. doi:10.1109/jiot.2018.2863555.

[13] Yu, Bin, Huaizhu Wang, Wenxuan Shan, and Baozhen Yao. "Prediction of Bus Travel Time Using Random Forests Based on Near Neighbors." Computer-Aided Civil and Infrastructure Engineering 33, no. 4 (November 1, 2017): 333–350. doi:10.1111/mice.12315.

[14] Petersen, Niklas Christoffer, Filipe Rodrigues, and Francisco Camara Pereira. "Multi-Output Bus Travel Time Prediction with Convolutional LSTM Neural Network." Expert Systems with Applications 120 (April 2019): 426–435. doi:10.1016/j.eswa.2018.11.028.

[15] Office of the Deputy Mayor for Planning and Economic Development "International Business|dmped." Available online: https://dmped.dc.gov/page/international-business (accessed on 13 July 2020).

[16] Washington Metropolitan Area Transit Authority "Metrobus|WMATA." Available online: https://www.wmata.com/service/bus/ (accessed 29 July 2020).

[17] U. svg: U. S. C. B. Lokal_Profil _w_territories svg:, English: This is a map of the location of the District of Columbia showing the borders of states and counties around it. 2010.

[18] E.|M., "Mean Absolute Error~MAE [Machine Learning (ML)]," Medium, Available online: https://medium.com/@ewuramaminka/mean-absolute-error-mae-machine-learning-ml-b9b4afc63077 (accessed on 14 June 2020).

[19] Contemporary Analysis, K. D. says, "Using Mean Absolute Error to Forecast Accuracy," Available online: https://canworksmart.com/using-mean-absolute-error-forecast-accuracy/ (accessed on 14 June 2020).