


Development and Evaluation of Recurrent Neural Network-Based Models for Hourly Traffic Volume and Annual Average Daily Traffic Prediction

Zadid Khan¹, Sakib Mahmud Khan¹, Kakan Dey², and Mashrur Chowdhury¹

Transportation Research Record
2019, Vol. 2673(7) 489–503
© National Academy of Sciences:
Transportation Research Board 2019
Article reuse guidelines:
sagepub.com/journals-permissions
DOI: 10.1177/0361198119849059
journals.sagepub.com/home/trr
 SAGE

Abstract

The prediction of high-resolution hourly traffic volumes of a given roadway is essential for transportation planning. Traditionally, automatic traffic recorders (ATR) are used to collect these hourly volume data. These large datasets are time-series data characterized by long-term temporal dependencies and missing values. Regarding the temporal dependencies, all roadways are characterized by seasonal variations that can be weekly, monthly or yearly, depending on the cause of the variation. Traditional time-series forecasting models perform poorly when they encounter missing data in the dataset. To address this limitation, robust, recurrent neural network (RNN)-based, multi-step-ahead forecasting models are developed for time-series in this study. The simple RNN, the gated recurrent unit (GRU) and the long short-term memory (LSTM) units are used to develop the forecasting models and evaluate their performance. Two approaches are used to address the missing value issue: masking and imputation, in conjunction with the RNN models. Six different imputation algorithms are then used to identify the best model. The analysis indicates that the LSTM model performs better than simple RNN and GRU models, and imputation performs better than masking to predict future traffic volume. Based on analysis using 92 ATRs, the LSTM-Median model is deemed the best model in all scenarios for hourly traffic volume and annual average daily traffic (AADT) prediction, with an average root mean squared error (RMSE) of 274 and mean absolute percentage error (MAPE) of 18.91% for hourly traffic volume prediction and average RMSE of 824 and MAPE of 2.10% for AADT prediction.

Hourly traffic volume roadway data comprise an important high-resolution dataset used to describe the operational characteristics of a transportation system. Accurate hourly traffic volumes can be utilized in calculating the annual average daily traffic (AADT). AADT is an essential parameter in many transportation models and decisions. Moreover, the prediction of future hourly traffic volumes of a given roadway is also important because it can be used to estimate future growth. Moreover, the high-resolution data provide insight into the factors contributing to growth, as there may be a gradual growth pattern or a sudden peak. The roadway volume can increase at a very specific time next year as a result of some special events, and a predictive model can predict this change. This means that the special event is a phenomenon that has occurred before. However, if a predictive model is unable to capture this event, then it is a new phenomenon that has not been previously observed. Therefore, the detection of special events or

anomalies is also an application of high-resolution hourly volumes.

Transportation planning is characterized by many projects that are related to future infrastructure investments (i.e., designing new roadways and bridges, urbanization/land development, the addition of new lanes in a roadway, new medians, and new traffic signals). As such, the expensive and time-consuming nature of these projects means that the project team must ensure the importance of a project before an investment is forthcoming. A predictive model that accurately forecasts hourly volumes and AADT can help in increasing confidence in the

¹Glenn Department of Civil Engineering, Clemson University, Clemson, SC

²Department of Civil and Environmental Engineering, West Virginia University, Morgantown, WV

Corresponding Author:

Address correspondence to Zadid Khan: mdzadik@clemson.edu

investment decisions. Moreover, many transportation software and models use future AADT as input data. For example, AADT serves as input into the Safety Analyst Software through which city planners can analyze crash data and improve traffic safety (1).

There are several ways to collect continuous hourly traffic volumes for a roadway, most notably using automatic traffic recorders (ATRs), which are permanent traffic count stations strategically located at selected locations. These ATRs collect hourly volumes and can collect real-time average speed data. The technology used in the ATR can vary according to data-collection requirements. The most popular technologies used in ATR are inductive loops, magnetic counters, radar sensors, and surveillance cameras. The ATRs usually send the real-time data to a Department of Transportation (DOT) traffic data center, where it is archived and made available for internal and external usage through an open portal. State DOTs also collect short-term counts at hundreds of locations to supplement the permanent count station data.

The hourly traffic volume data from an ATR is a classic example of a time-series with multiple sub-patterns. The hourly volume counts for a day exhibit hourly count variation by the time of the day, as the volumes at off-peak hours (e.g., nighttime) are always less than during the day. Weekday and weekend variations also characterize such systems, with weekdays subject to a higher volume than weekends, or vice versa, depending upon the location. The volumes also vary from month to month depending on weather, seasonal events, and other external factors. Different changes to the roadway such as a temporary shutdown of lanes and special events can create random variation in the time-series, which may or may not be repeated every year. Finally, the ATR stations can be temporarily unavailable because of faulty equipment, and disruption in communication with the traffic data center.

The objective of this study is prediction of hourly volume for 365 days of the following year so that an accurate estimation of the AADT of the next year can be derived from the data of the previous years. Moreover, given that the high-resolution hourly volume prediction should capture all the hourly, weekly, and monthly variations that are present in the dataset from previous years, a predictive model can be developed for multi-step-ahead forecasting.

This study details the development of such a model, the recurrent neural network (RNN) based time-series forecasting model with missing data treatment for predicting hourly traffic volume and AADT. RNN models are specialized deep learning models for sequence prediction that capture the different patterns in previous time steps and make a reliable estimate of future time steps

(2). Unfortunately, the missing data is a major problem in time-series because they are part of the sequence and cannot be discarded from the dataset. Moreover, hourly volumes for every hour in the input dataset are required to calculate an accurate estimate of AADT from the ATR data. Given that the removal of missing data will greatly reduce the prediction accuracy, a solution for this missing data input is required. In this paper, the authors describe the two approaches used, combined with the RNN, to address this issue. The architecture of the general predictive model is presented, including the missing data treatment layer and the RNN layer. Different variations of the proposed model are evaluated to identify the best model accordingly.

Literature Review

Based on the context of this research, the literature review has been divided into three segments:

- Missing data in prediction
- Time-series prediction with RNN
- Traffic volume and AADT prediction

Missing Data in Prediction

An incomplete dataset is always a challenge for time-series forecasting. However, imputing missing values in the historical data can provide a more robust prediction model. Comparative studies have been performed on various imputation methods for traffic data (3) and health survey data (4). Che et al. developed GRU-D from the basic gated recurrent unit (GRU) concept to mask the missing data in the historical database (5). They used the prediction model on synthetic and real-world clinical datasets. The GRU-D model exhibited a similar complexity and computation time compared with original RNN models. The authors also compared with basic RNN models coupled with other baseline imputation methods such as expectation maximization, principal component analysis, k -nearest neighbor, and Softimpute. The authors found that GRU-D model performed better than other imputation methods. Similarly, in their use of pattern mixture kernel submodels (PMKS), which includes submodels for each missing data pattern, Anava et al. found that PMKS outperformed different imputation models and complete-case submodels (6).

Time-Series Prediction with RNN

Time-series analysis is a major field of research. The state-of-the-art approaches to time-series prediction have been discussed by Chatfield (7). Deep learning models

have been applied to traffic data prediction and it has revealed their potential to make high-accuracy prediction. Långkvist et al. conducted a comparative study about unsupervised and supervised deep learning models for time-series prediction (8). The RNN model is one of the popular methods for sequence prediction. Internal feedback connections with the hidden RNN layers have been used to model the temporal relationship within the time-series data explicitly. Based on the actual output and the predicted output, the error is calculated, and the weights of the networks are revised until the model converges. RNN, which is also notable for a high prediction accuracy for noisy datasets, has been successfully used in financial forecasting (9), health care (5), transportation, chaotic time-series prediction (10), and acoustic modeling (11). In that regard, Giles et al. applied RNN in financial forecasting using financial datasets that are small, noisy, and non-linear (9). Using RNN, they combined symbolic methods with RNN to overcome the issues related to the unequal a-priori class probabilities and overfitting. Also, by rejecting data having low confidence measure, the prediction model achieved 40% less error. In their use of long short-term memory (LSTM) to predict a large-scale acoustic model, Sak et al. noted that the five-layered LSTM RNN outperformed one, two, three, and seven-layered LSTM RNN networks (11). These networks were trained with a hand-transcribed and anonymized dataset with three million distinct pieces of data.

For multi-step prediction (i.e., predicting for a time-series sequence), Han et al. used RNN in which nodes are operated non-linearly, the output of which was linked with the input of that node and the following node (10). A self-adaptive and extended Kalman filter-based back-propagation method was used to achieve a superior level of performance in the extended method [i.e., root mean squared error (RMSE) of 3.9 for 6 min lag time] compared with the standard RNN model (i.e., RMSE 4.4 for a 6 min lag time). However, the increase of time horizon for the multi-step prediction caused a decrease in the prediction accuracy of all models. Bao et al. reported similar findings regarding the multi-stage predicted value of the current time step serving as the input for the next time step (12). A single-step RNN model outperformed the multi-stage RNN model, the linear regression model, and the hidden Markov model.

Traffic Volume and AADT Prediction

Previous research has been successful in applying deep learning to the traffic flow prediction problem. Lv et al. developed a stacked autoencoder-based model (13), and Hunag et al. developed deep belief networks with multi-task learning (14). For predicting short-term traffic flow,

Tian and Pan used an LSTM model with dynamically optimal time lag to predict short-term traffic (15). To determine the optimal time lag in real-time, the LSTM RNN was used with memory blocks of three multiplicative units. Thirty loop detection datasets from six California freeways were used (from California PeMS Database). The data were aggregated at 15, 30, 45, and 60-min intervals. The dynamic time lag was then used to derive the lowest mean absolute percentage error (MAPE) of 6.5% compared with the support vector machine, the random walk, the single layer feed-forward neural network, and the stacked autoencoder models. Similarly, in Fu et al. (16), LSTM (mean absolute error or MAE 18%) and the GRU neural network (MAE 17%) were applied to California PeMS data to predict an improved short-term traffic flow over the autoregressive integrated moving average (ARIMA) model (MAE 19%). To predict AADT, Castro-Neto et al. used support vector regression (SVR), Holt's exponential smoothing (HES), and ordinary least-square linear regression (OLS-regression) (17). Specifically, in their use of a 20-year AADT dataset from Tennessee, they noted that SVR (MAPE 2.3%) outperformed both the Holt-ES (MAPE 2.7%) and OLS-regression (MAPE 3.9%). Another study conducted by the Idaho Department of Transportation noted that the classification and regression trees (CART) method was more accurate than growth factor, linear regression, and multiple regression methods (18).

Method

In this section, the three-step research method is outlined as shown in Figure 1; each step is described in a separate subsection.

Step 1: Data Collection and Pre-Processing

Before developing the model, ATR data from 163 ATRs at different locations across the state of South Carolina were collected for past 10 years (2008–2017). These data were then classified into seven functional ATR highway groups. The names and counts of each group are provided below:

- Rural interstate (38 ATRs)
- Urban interstate (45 ATRs)
- Rural arterial (37 ATRs)
- Urban arterial (22 ATRs)
- Rural collector (9 ATRs)
- Urban collector (6 ATRs)
- Local roads (6 ATRs)

Of these functional groups, the last group (local roads) only has six ATRs, and all six ATRs were established in

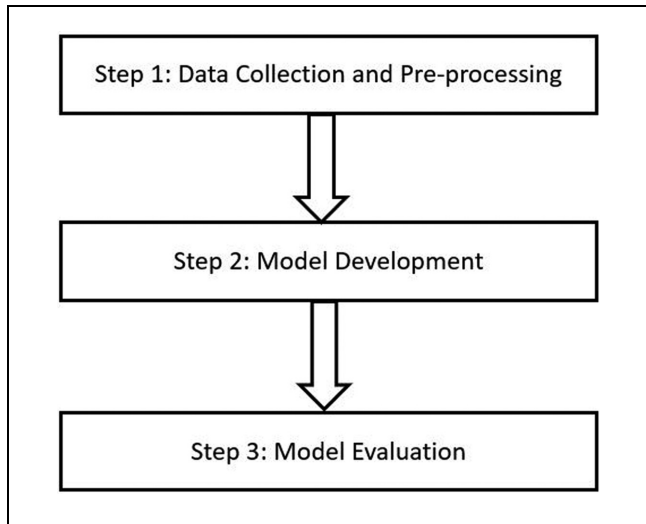


Figure 1. Steps of the research method.

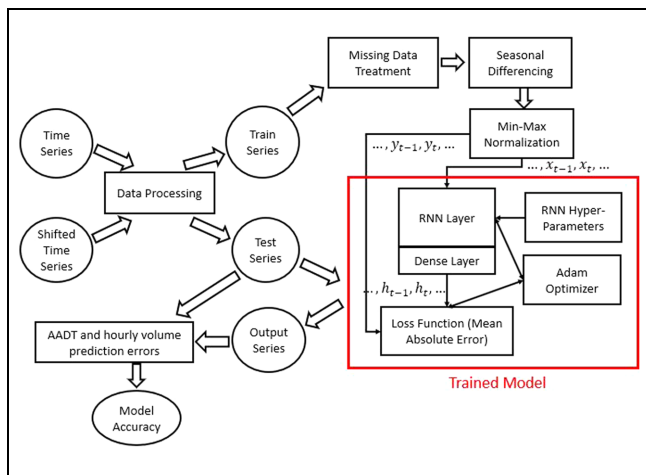


Figure 2. Block diagram of hourly volume (and AADT) prediction model.

Note: AADT = annual average daily traffic; RNN = recurrent neural network.

2013, so only 5 years of data are available for local roads. Therefore, this study uses the first six ATR categories from the first six functional groups. Although the data are openly available on the South Carolina DOT website, they are not stored in a structured database or a file. Python was used to create a script to collect and store the web interface data into text files. Data from all ATRs for 10 years were collected, and for each ATR, the hourly volume time-series contains 87,672 h of data. However, some ATRs have a percentage of missing value greater than 20% over the 10-year period of 2008–2017. Only 92 ATRs out of the 163 ATRs have missing value less than 20%. Only these ATRs are considered throughout the analysis.

After collection, data were prepared for model input to meet the goal of predicting the volume time-series 365 days (a full year) in advance. Therefore, the authors performed a forward time shift of the data by $365 \times 24 = 8,760$ h and created a second time-series. The original time-series serves as the input and the shifted time-series serves as the output. Both data series were augmented to form the prediction dataset. As a result of the forward shift operation, the number of data points in the prediction dataset was 78,912. Both the input and the output time-series may contain missing data in the form of “NaN” (Not a Number) values, which are left unchanged in the dataset. “NaN” is a numeric data type used to fill empty spaces in a numeric vector.

Step 2: Model Development

The block diagram of the 365-day ahead hourly volume prediction model is detailed in Figure 2. The components of the diagram are described below in detail.

Data Processing. The two time-series were first combined and then separated into the training and testing datasets. AADT prediction was done for the last 2 years (2016 and 2017) of the dataset. The total number of days in 2016 and 2017 was 731. Therefore, the total number of hours in the testing set was 731×24 , or 17,544, thus leaving the training set with 61,368 h of data.

Missing Data Treatment. To overcome the missing data problem, masking and imputation were used, both of which are described below.

The missing data were masked via a masking layer, the function of which is replacing all missing values with a specific value in the dataset. In the training dataset, the missing values are represented as “NaN.” The “NaN” values are then replaced by a masking value, which is set at “-1” here. The model learns that these “-1”s are missing values, so whenever it finds a “-1” in the testing set, it predicts accordingly. The reason for masking with “-1” is to differentiate between actual data and missing data. “-1” will never appear naturally in the hourly volume dataset because this is not a valid hourly volume value. Thus, “-1” is a good masking value.

The second approach adopted was the data imputation. The function of the imputation layer is estimating the missing values for a variable based on the existing data available for other time steps and other variables. Although several popular imputation methods are used with time-series data, the five below were selected, all of which were implemented in Python.

Mean/median imputation is the simplest form of imputation method, in which the authors separated the dataset from the missing values, took the mean or the

median of the separated dataset, and replaced the missing values with this value in the original dataset. Regarding computation complexity, this is a simple and computationally inexpensive method.

The expectation maximization (EM) method is used for many different applications, particularly data imputation. The core idea of this algorithm is using the available data to predict the missing data in a dataset (19). Let us assume that the set of missing values is Z and the rest of the values form the set X . Using the value X , a model can then be developed with the parameter set of θ , and then the model can be applied to estimate the values in Z . However, given that both θ and Z are initially unknown quantities, the method will be an iterative process that will converge on a solution.

The EM algorithm has two steps, expectation step and maximization step. In the expectation (E) step, the algorithm assumes that θ is known and calculates the expected value of the log-likelihood (LogL) function (Equation 1). The likelihood function calculates a probability value for each point being in a certain subgroup of the overall solution, defined by a subset of the parameter set θ . In the maximization (M) step, the algorithm finds the optimum parameter that maximizes the log-likelihood function (Equation 2). Maximizing this function ensures that θ is converging toward the correct values:

$$Q(\theta|\theta') = E_{Z|X, \theta'}[\log L(\theta; X, Z)] \quad (1)$$

$$\theta^{t+1} = \operatorname{argmax}_{\theta} Q(\theta|\theta') \quad (2)$$

Here, Q is the expectation value, θ' refers to value of the parameter set θ at time step t .

Multivariate imputation by chained equations (MICE) is a popular method for data imputation. It is used when the dataset has multiple variables. In the first step, all missing values are replaced with a placeholder value (in this study, the mean was used). Afterward, the algorithm assumes one variable as the dependent variable and all other variables as independent variables. The algorithm performs regression (this study used linear regression) to calculate the missing values for the dependent variables. The same process is performed for all other variables through several cycles until satisfactory results are achieved (20).

k -nearest neighbor (KNN) is a non-parametric algorithm to impute the missing values in the dataset. The k -nearest neighbor can function for any dataset as long a relationship between adjacent indices of the dataset exists, which is true for the hourly volume time-series data. The algorithm calculates the Euclidian distances from the k -nearest neighbors of the missing value and then estimates the missing value as the average of the k -nearest values (21). As its

application here is to a time-series and the short-term dependency is significant, a small k value of 5 was chosen in this study.

The random forest (RF) algorithm is an extension of the decision tree algorithm. First, it separates the missing data from the actual dataset. It then randomly samples a subset of data points from the clean dataset during multiple iterations to create numerous decision trees using each subset. Finally, the missing value from all decision trees is calculated to derive the average of the output (22).

Seasonal Differencing and Min–Max Normalization. After solving the missing data problem, seasonal differencing was applied on the entire training dataset, and then the Min–Max normalization was applied. The hourly volume time-series for ATRs is a non-stationary time-series, which has different trends and seasonal components. A stationary time-series is a series whose mean, variance, and autocorrelation structures do not change over time. In other words, the time-series will not have any time-varying trend or seasonal components. The traffic volumes usually increase over time, and the growth may contain linear or exponential trend or seasonal components. The non-stationary time-series as input will result in inaccurate predictions from the RNN model. To avoid this scenario, seasonal differencing was used on the data so that the time-series becomes stationary and the statistical properties remain unchanged for the whole duration. Seasonal differencing can be defined as the process that creates a time-series that includes the changes from season to season. For seasonal differencing, a lag version of the time-series is subtracted from the original time-series. The interval of time lag needs to be identified for differencing. As the time-series is based on hourly data, the authors have identified that the optimum interval that works well for all ATRs is 8,736 (52 weeks = 364 days = 8,736 h). Afterward, the data were normalized between 0 and 1. After forecasting using RNN/GRU/LSTM models, the forecasted data were inverted to get back to the original condition. Equations 3 and 4 describe the functions used in this study to standardize the dataset:

$$x'_t = (x_t - x_{t-8736}) \quad (3)$$

$$x''_t = \frac{x'_t - x_{\min}}{x_{\max} - x_{\min}} \quad (4)$$

Here, x'_t is the output of the seasonal differencing, x_t is the input time-series and x_{t-8736} is the lagged time-series by 8736 hours. x''_t is the normalized time-series and x_{\min} and x_{\max} are the minimum and maximum values of the seasonal differenced time-series x'_t respectively.

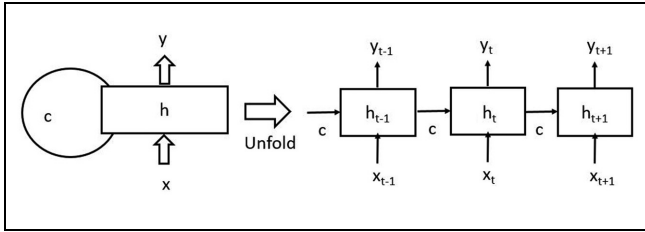


Figure 3. Basic recurrent neural network (RNN).

Baseline Methods. Before describing the RNN models, the baseline methods need to be discussed. In this study, the authors used three baseline models for comparison with RNN-based models. The three baseline models were linear regression, ARIMA, and HES models. These are popular models for time-series forecasting. The linear regression model identifies a straight line to fit the time-series based on least-square principles. The regression model was implemented using the “Scikit-learn” package in Python (23). ARIMA and HES models are briefly described below.

ARIMA stands for autoregressive integrated moving average. It is a moving average model that also uses seasonal differencing, and forecasts for future time steps using some number of previous time steps in the dataset. This model has three parameters, the lag order, the degree of differencing, and the size of the moving average (MA) window. As seasonal differencing is being applied to the time-series to make it stationary, the differencing parameter is not required, so it was set to zero in this study. The lag order and MA window size for each ATR’s ARIMA model were identified using a grid search method. The values were extracted from the model with the least Akaike Information Criterion (AIC) value (24). AIC is used to measure the quality of a statistical model compared with other models. The ARIMA model was implemented using the “Statsmodels” package in Python (25).

HES stands for Holt’s exponential smoothing. It is also known as basic/single exponential smoothing technique (EST). This smoothing technique can be best explained using Equation 5:

$$y_t = \alpha \left(x_t + \sum_{i=1}^{t-1} (1 - \alpha)^i x_{t-i} \right) + (1 - \alpha)^t x_0 \quad (5)$$

From Equation 5, it can be observed that each new prediction (y_t) depends on all the previous values (x_t) in the time-series with continuously increasing powers of coefficients, thus it is called exponential smoothing. Here α is the smoothing factor and it can be anywhere between 0 and 1. A value closer to 1 gives more importance to recent observations in time-series, whereas a value closer to 0 gives more importance to smoothing (26). The HES

model was also implemented using the “Statsmodels” package in Python (25).

Recurrent Neural Networks. The predictive model is based on a single RNN layer followed by a dense layer with a single neuron. The authors found that this simple architecture performed the best for this dataset. The RNN layer is an individual computation block where the current output is fed back as input for the next step. The structure of a basic RNN model is illustrated using Figure 3. Here, x is the input, y is the output, h is the transfer function and c is the cell state.

Let us assume that x_t is the input, y_t is the output, h_t is the transfer function of the repeater block at time t , and c is the internal loop in the cell. As a basic RNN cell unfolds, it assumes the form of a chain of cells characterized by a cell-to-cell transfer of vector c . As such, the cell “memorizes” the previous output when computing the next. In this study, three types of RNN model were implemented: simple RNN, GRU, and LSTM. The difference in the RNN models lies in the transfer function (h) of the repeater block in Figure 3. The description of each is given below.

In the simple RNN model, the transfer function h is merely an activation function. In this study, the authors used the “tanh” activation function, as shown in Figure 4. The “tanh” activation function ensures that the output is normalized between -1 and 1 . The simple RNN model operates on two Equations, Equation 6 and 7. The input to the model is x_t and c_{t-1} , and the output of the model is y_t and c_t :

$$y_t = \tanh(W_y x_t + U_y c_{t-1} + b_y) \quad (6)$$

$$c_t = \tanh(W_c x_t + U_c c_{t-1} + b_y) \quad (7)$$

The simple RNN model is characterized by the vanishing gradient problem, preventing the model from determining the long-term internal dependencies. The continuously decreasing gradient of the previous time steps is equivalent to the network forgetting about those time steps. The LSTM model solves the vanishing gradient problem by introducing some additional operations, as shown in Figure 4 (27). The top horizontal line, known as the cell state (c), is the memory of the LSTM model, where pointwise additive and multiplicative operations are performed to add or remove information from memory. These operations are performed using the input and forget gates of the LSTM block, which also contains a “tanh” activation function for the output. Equations 8–12 describe the mathematical operations inside the LSTM neurons. The input to the model is x_t , c_{t-1} and y_{t-1} . The output is y_t and c_t . σ indicates sigmoid activation function. Intermediate terms generated inside the neuron are f_t , i_t , and o_t , which correspond to the output

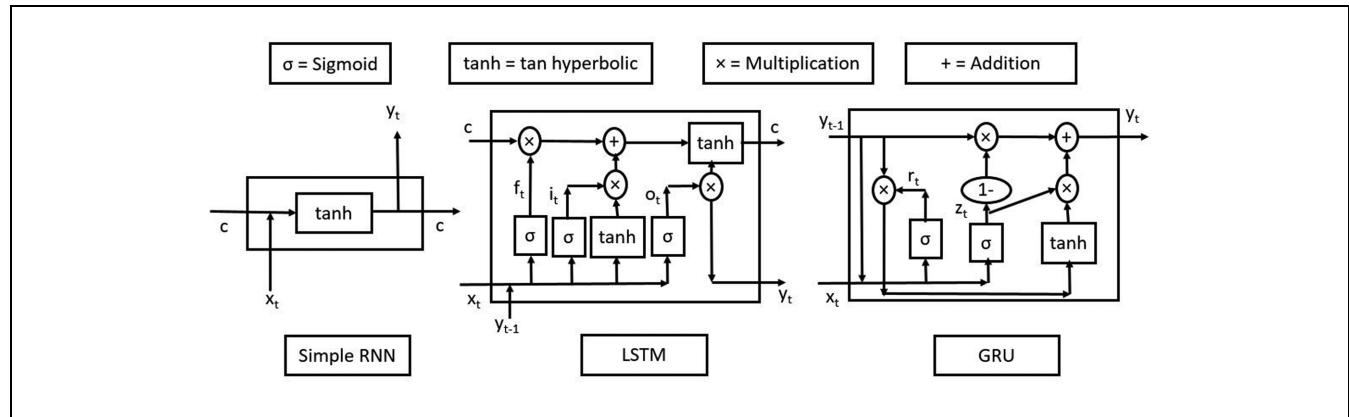


Figure 4. Repeater block transfer function for each model.

Note: RNN = recurrent neural network; GRU = gated recurrent unit; LSTM = long short-term memory.

of the forget, input, and output gates, respectively. W , U , and b are the weights corresponding to the three gates in the neuron. The dots (.) in the equation indicate elementwise multiplication instead of matrix multiplication. All the terms in the equations can be found in Figure 4:

$$f_t = \sigma(W_f x_t + U_f y_{t-1} + b_f) \quad (8)$$

$$i_t = \sigma(W_i x_t + U_i y_{t-1} + b_i) \quad (9)$$

$$o_t = \sigma(W_o x_t + U_o y_{t-1} + b_o) \quad (10)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh(W_c x_t + U_c y_{t-1} + b_c) \quad (11)$$

$$y_t = o_t \cdot \tanh(c_t) \quad (12)$$

The GRU is a simplified version of the more complex LSTM unit. It combines the input and forgets gates into a single update gate. It then merges both the cell and hidden states for faster operation (28). Equations 13–15 describe the mathematical operations inside the GRU neurons. The intermediate states of the GRU neuron are z_t and r_t . The terms in the equations can be found in Figure 4:

$$z_t = \sigma(W_z x_t + U_z y_{t-1} + b_z) \quad (13)$$

$$r_t = \sigma(W_r x_t + U_r y_{t-1} + b_r) \quad (14)$$

$$y_t = z_t \cdot y_{t-1} + (1 - z_t) \cdot \tanh(W_y x_t + U_y (y_{t-1} \cdot r_t) + b_y) \quad (15)$$

The basic diagram of GRU is shown in Figure 4. A dense layer with one neuron follows up the RNN layer. Here a loss function and a gradient descent optimizer were used to find the optimum weights for this network. All the models were implemented using Keras (29) in Python and a Tensorflow (30) backend.

Step 3: Model Evaluation

After training the model, it predicts the output, and then the output is compared with the ground truth by calculating the errors for hourly traffic volume and AADT prediction. The error block segments the data into blocks of 365 or 366 days based on the prediction year, sums all the hourly volumes, and then divides the sum by the total number of days. Two types of error measures were used to compare the models, RMSE and MAPE. The performance of different models is detailed in the “Analysis and Results” section below.

Analysis and Results

The analysis and results section contains multiple steps. The steps are described below.

At first, the data from six sample ATRs of six different functional classes were used to develop the models. The ATRs selected for analysis are ATR #1, 6, 15, 21, 39, and 41. The ATRs were selected randomly from each functional class. For each ATR, the data from 2008 to 2017 were collected and the model was trained using the data from 2008 to 2015. The models were then evaluated using their accuracy to predict the AADT of 2016 and 2017. Model accuracy is the opposite measure of the MAPE values. Equation 16 represents the formula of accuracy of the model. For each ATR, 21 different models were compared based on the type of missing data treatment method and the variety of transfer functions used in the repeater block of the RNN model. Based on the accuracy, the best model is selected. The model is checked for overfitting/underfitting issues:

$$\text{Accuracy} = (100 - \text{MAPE})\% \quad (16)$$

The best model is compared with the three baseline methods (linear regression, ARIMA, HES) for 92 ATRs.

These four models were used to predict the hourly traffic volumes and AADT of 2016 and 2017 for 92 ATRs. The mean and variance of RMSE and MAPE values are reported for different functional classes.

A statistical significance test was performed for AADT prediction of 92 ATRs using the four models. The test indicates if there is any significant difference between the actual mean and variance compared with model mean and variance.

Next, the efficacy of the best model in accurately capturing the trends in the time-series between 2016 and 2017 was analyzed. A visual representation is shown on how the model captures the long-term dependencies. A sample ATR is chosen for illustration, which in this case is ATR #39. The visual comparison is accompanied by the RMSE and MAPE values of hourly volume prediction and AADT prediction.

Finally, instead of 1-year look-ahead, a multi-year look-ahead prediction (2-year ahead and 3-year ahead) was performed to verify the performance of the best model. This step focused on only AADT prediction, and the experiment was conducted using a sample ATR, which in this case is ATR #21.

RNN Model Hyper-Parameters

For each RNN/GRU/LSTM model, the hyper-parameters were varied to identify the best parameter for each model. For each model, two hyper-parameters were varied: the number of epochs and batch size. The values of these hyper-parameters were identified from the model with the lowest validation error. Other hyper-parameters of the models were constant for all ATRs in this analysis. The values of the hyper-parameters used in this study are given below:

- Optimizer: Adam (31), Learning Rate: 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$
- Loss function: Mean Absolute Error
- Number of hidden layers of RNN/GRU/LSTM: 1
- Number of neurons in hidden layer: 1
- Dropout was not required since there was no overfitting / underfitting issues

Comparison of Different RNN Models

Rural Interstate. First, a thorough analysis was performed using a sample ATR located in a rural interstate; in this case it is ATR #15. The results for ATR #15 for the years 2016 and 2017 are shown in Figures 5 and 6, respectively. The LSTM-Median model has the highest accuracy in 2016, and the LSTM-MICE model has the highest accuracy in 2017. The accuracy of the best model is highlighted with a circle.

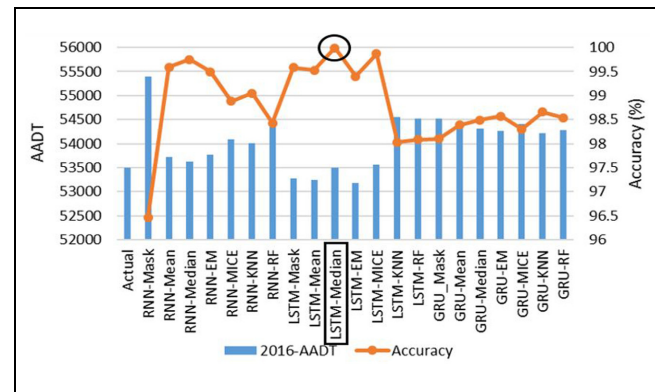


Figure 5. Prediction results for ATR #15 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

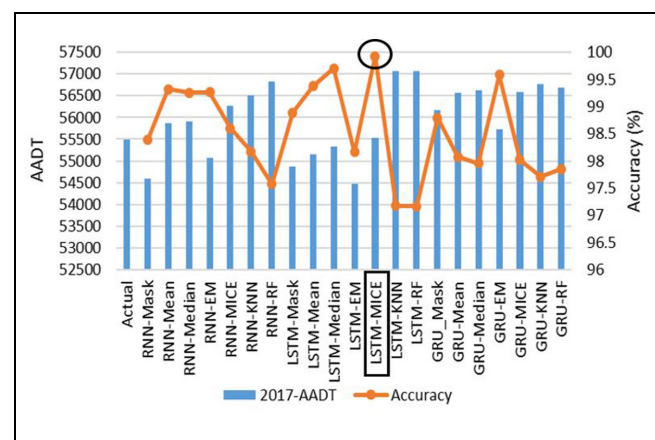


Figure 6. Prediction results for ATR #15 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

Urban Interstate. For urban interstate, ATR #21 was chosen. The AADT predictions of ATR #21 for the years 2016 and 2017 are shown in Figures 7 and 8, respectively. The LSTM-Median model has the highest accuracy for 2016 and RNN-RF model has the highest accuracy for 2017.

Rural Arterial. For rural arterial, ATR #1 was chosen. The AADT predictions of ATR #1 for the years 2016 and 2017 are shown in Figures 9 and 10, respectively. The LSTM-Mean model has the highest accuracy for both 2016 and 2017.

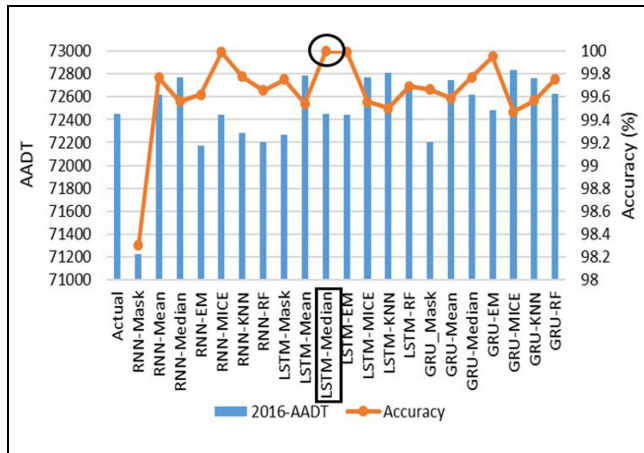


Figure 7. Prediction results for ATR #21 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

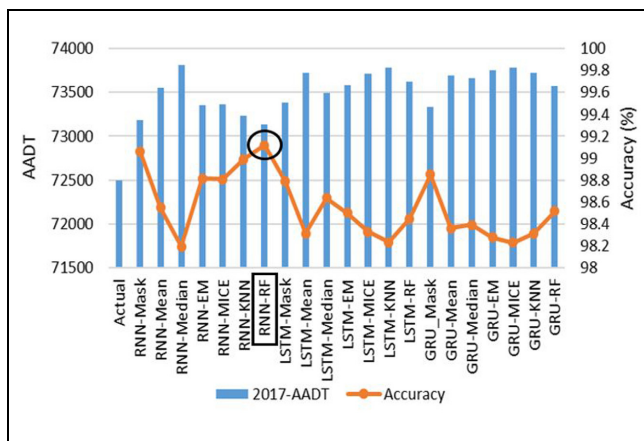


Figure 8. Prediction results for ATR #21 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

Urban Arterial. For urban arterial, ATR #6 was chosen. The AADT predictions of ATR # 6 for the years 2016 and 2017 are shown in Figures 11 and 12, respectively. The GRU-Mean model has the highest accuracy for 2016 and LSTM-KNN model has the highest accuracy for 2017.

Rural Collector. For rural collector, ATR #39 was chosen. The AADT predictions of ATR #39 for the years 2016 and 2017 are shown in Figures 13 and 14, respectively. The

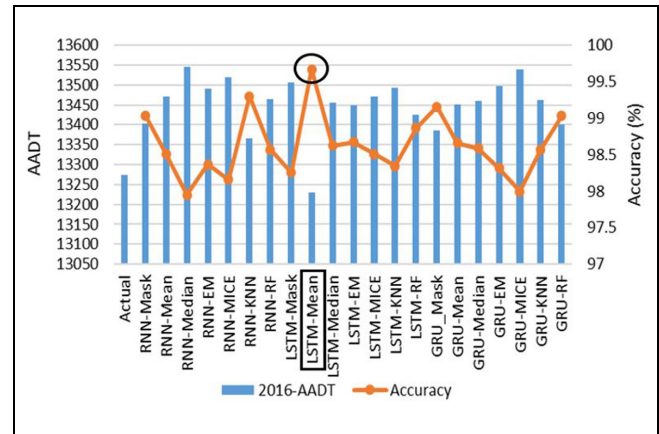


Figure 9. Prediction results for ATR #1 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

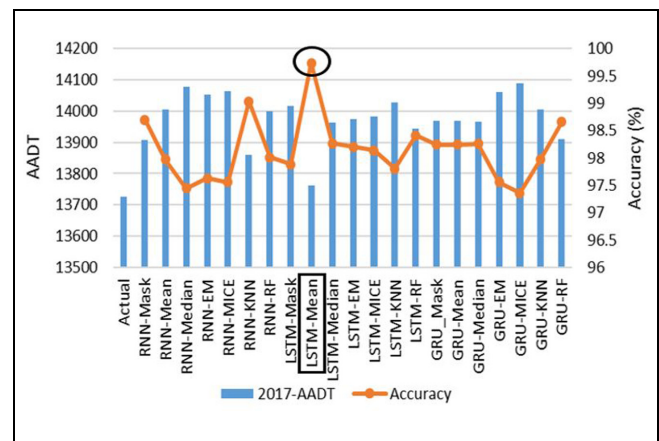


Figure 10. Prediction results for ATR #1 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

model RNN-KNN has the highest accuracy for 2016 and LSTM-Median model has the highest accuracy for 2017.

Urban Collector. For urban collector, ATR #41 was chosen. The AADT predictions of ATR #41 for the years 2016 and 2017 are shown in Figures 15 and 16, respectively. The LSTM-KNN model has the highest accuracy for 2016 and LSTM-EM model has the highest accuracy for 2017.

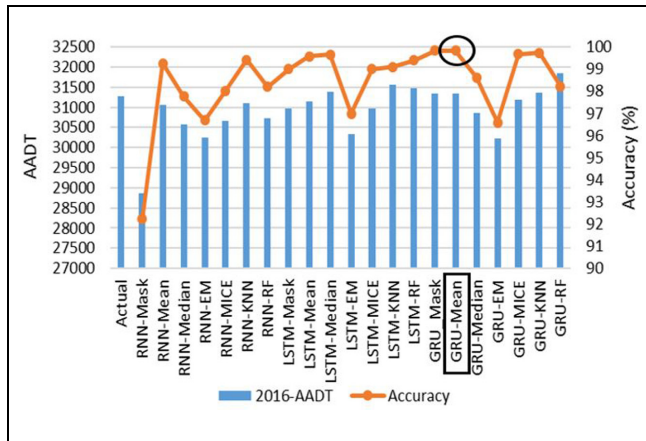


Figure 11. Prediction results for ATR #6 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

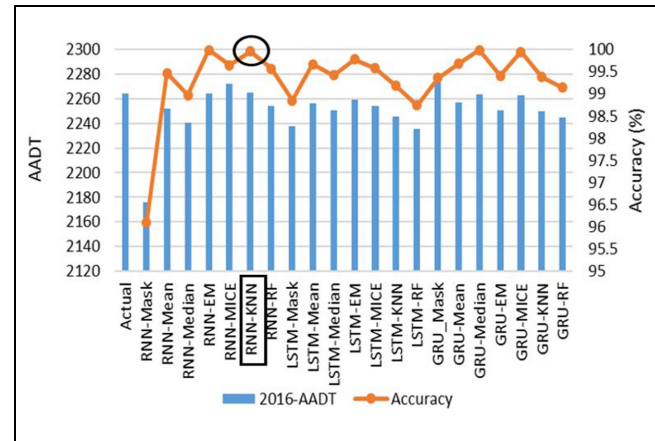


Figure 13. Prediction results for ATR #39 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

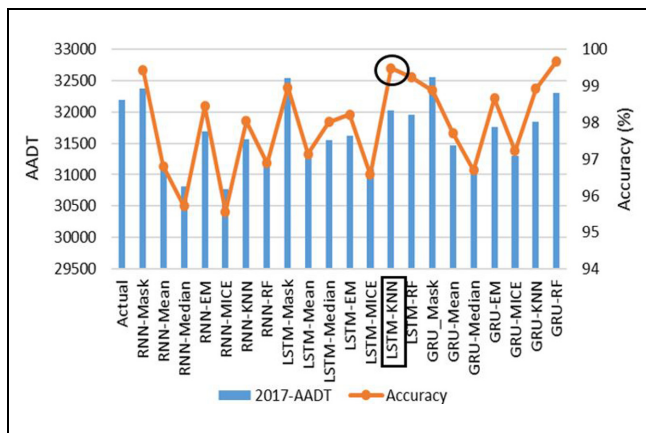


Figure 12. Prediction results for ATR #6 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

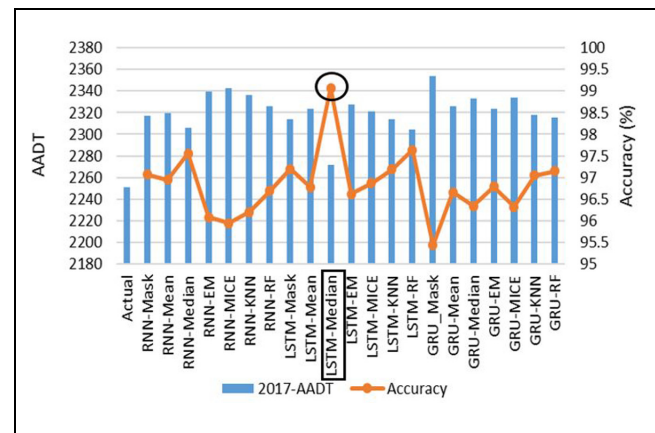


Figure 14. Prediction results for ATR #39 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR= automatic traffic recorders.

Comparison Summary. In the analysis of the performance of different models, all RNN-based models show comparable performance based on average RMSE and MAPE of AADT prediction. The LSTM-Median model is chosen for all roadway functional groups for its overall superior performance compared to other models considered in this study. The performance of the LSTM model indicates a significant dependency of the hourly volume on the past data, which the simple RNN model is unable to capture properly. Different imputation methods work well for different functional groups, and imputation

performs better than masking in all cases, suggesting that the model struggles to learn from the missing data. Consequently, imputation rather than masking is the preferred method. The performance of the imputation methods depends on the extent of missing data. The percentages of missing values for the ATRs are given in Table 1. Because of the higher percentage of missing values in ATR #6 and ATR #41, the more sophisticated KNN method and EM method work better. Concerning the lower percentage of missing values in ATR #1, #15, #21, #39, however, the simple imputation methods of mean

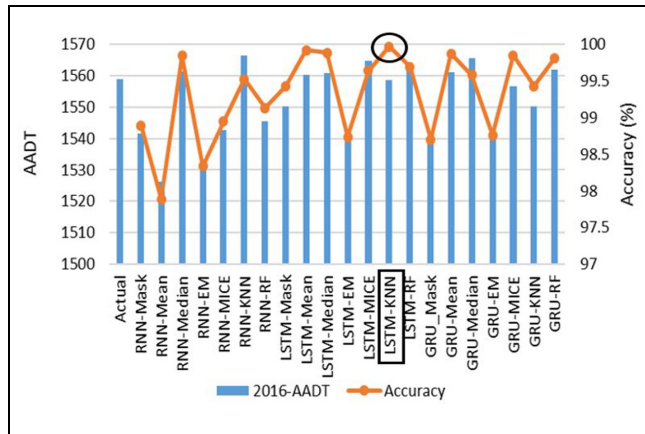


Figure 15. Prediction results for ATR #41 in 2016.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

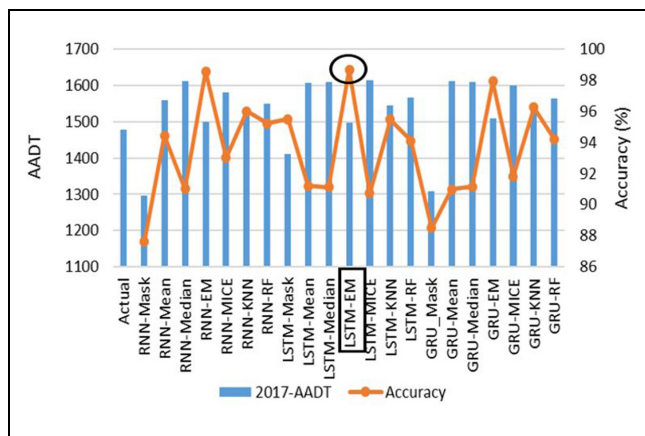


Figure 16. Prediction results for ATR #41 in 2017.

Note: AADT= annual average daily traffic; RNN = recurrent neural network; EM = expectation maximization; MICE = multivariate imputation by chained equations; KNN = k-nearest neighbor; RF = random forest; LSTM = long short-term memory; GRU = gated recurrent unit; ATR = automatic traffic recorders.

and median perform better. This finding suggests that imputing the missing values with mean/median is a reasonable method if the percentage of missing values is low.

Overfitting/Underfitting Issues

Based on the analysis in the previous section, the LSTM-Median model is superior with respect to other models. However, it should be verified that there are no overfitting or underfitting issues with the model. Therefore, the loss function (MAE) values of the LSTM-Median model

Table 1. Missing Data (%)

ATR	Missing data (%)
1	1.3
6	4.9
15	2.2
21	2.9
39	2.1
41	6.7

Note: ATR = automatic traffic recorders.

Table 2. Training and Validation MAE

ATR #	Training MAE	Validation MAE	Overfit (%)
1	0.020	0.023	15%
6	0.032	0.033	3%
15	0.029	0.036	24%
21	0.021	0.025	19%
39	0.026	0.025	−4%
41	0.042	0.052	24%

Note: ATR = automatic traffic recorders; MAE = mean absolute error.

for training and validation set for the six ATRs which have been analyzed in this study are checked. The values of the loss MAEs are given in Table 2. As it can be observed, the overfit (i.e., $\frac{\text{Training MAE} - \text{Validation MAE}}{\text{Training MAE}} \times 100\%$) does not exceed 25% in any case. For ATR #39, the model actually has a better validation MAE value compared with training MAE, thus the negative sign in the overfit percentage. The maximum overfit is 24% for ATR #15 and ATR #41.

Comparison with Baseline Models

As the LSTM-Median model was selected as the best model, a comparative study of this model was performed with the three baseline models: regression, ARIMA, and HES models. This study predicted the hourly traffic volume and AADT using all four models for 92 ATRs. Then, the RMSE and MAPE values were calculated for the 92 ATRs using all four models. The mean and standard deviation (SD) of RMSE and MAPE values were then calculated. As mentioned previously in the method section, ATRs with greater than 20% missing value over the 10-year period of 2008–2017 are not considered in the analysis. Out of 163 ATRs, only 92 ATRs have a missing value percentage less than 20%. The mean of the missing value percentages is 9.03% and the SD of the missing value percentages is 8.26% for these 92 ATRs. Table 3 contains all the results for the comparison of LSTM-Median model and the baseline models.

Table 3. RMSE and MAPE Values of Baseline and LSTM Model

Baseline	Model	Statistical measure	Hourly volume RMSE	Hourly volume MAPE (%)	AADT RMSE	AADT MAPE (%)
Yes	Regression	Mean	912	157.03	1800	5.62
		SD	704	82.07	1517	3.43
Yes	ARIMA	Mean	412	25.21	1188	3.45
		SD	324	10.37	1191	2.52
Yes	HES	Mean	414	29.70	1337	3.50
		SD	326	13.15	1745	3.07
No	LSTM	Mean	274	18.91	824	2.10
		SD	216	6.55	933	2.05

Note: RMSE = root mean square error; MAPE = mean absolute percentage error; LSTM = long short-term memory; AADT = annual average daily traffic; ARIMA = autoregressive integrated moving average; HES = Holt's exponential smoothing; SD = standard deviation.

From Table 3, it can be observed that the LSTM model has the least RMSE and MAPE for both hourly traffic volume and AADT prediction. Linear regression has the highest RMSE and MAPE values, as it is the simplest model. Because of the high mean (9.03%) and SD (8.26%) of missing values in different ATRs, the mean and SD of RMSE and MAPE values of different models have high values for hourly traffic volume prediction. Moreover, the missing values are responsible for high RMSE and MAPE values for hourly traffic volume prediction. The LSTM-Median model has an average accuracy of 81.09% for hourly traffic volume prediction and 97.9% for AADT prediction. The best baseline model is the ARIMA model, which has an average accuracy of 74.79% for hourly traffic volume prediction and 96.55% for AADT prediction. The developed model has improved the prediction accuracy from the baseline models.

Statistical Significance Test of Difference

In the previous subsection it was shown that, based on RMSE and MAPE values, the LSTM-Median model is superior compared with the baseline models. However, to prove that there is no significant difference between the model prediction and actual value, a statistical significance test was performed for all four models. The AADT for 2016 and 2017 was calculated for 92 ATRs using four models. First, an *F*-test was performed to identify if the variances are equal. Then, a *t*-test was performed to identify if the sample means are equal. The results of the statistical tests are given in Table 4.

From Table 4, it can be observed that all four models predict AADT with no significant difference with the actual values. However, the RMSE and MAPE values indicate that the LSTM model is better compared with other models. For example, the regression model may have reasonable accuracy in AADT prediction, but it is not able to capture the high-resolution seasonal variations in the traffic volume dataset. Therefore, the LSTM model is always better than the regression model.

Table 4. Test of Significant Difference

Model	<i>F</i> -test		<i>t</i> -test	
	<i>p</i> -value	Diff. in variance	<i>p</i> -value	Diff. in mean
Regression	0.53	Insignificant	0.69	Insignificant
ARIMA	0.48	Insignificant	0.92	Insignificant
HES	0.53	Insignificant	0.88	Insignificant
LSTM	0.51	Insignificant	0.94	Insignificant

Note: ARIMA = autoregressive integrated moving average; HES = Holt's exponential smoothing; LSTM = long short-term memory.

Long-Term Dependencies

Regarding hourly traffic volume prediction, the LSTM-Median model performs better than baseline models with an average MAPE of 18.91%. The authors next analyzed the hourly volume prediction and the model's ability to track the variations in the time-series. The actual hourly trend and the predicted hourly trend were compared using the LSTM-Median model for ATR #39. The comparison is shown in Figure 17.

The plot on the right side in Figure 17 shows five distinct transitions (highlighted with square boxes) in the actual hourly volume trend. The plot on the left side in Figure 17 shows the effect of the LSTM-Median model in capturing all five transitions (also highlighted with square boxes) accurately. These are seasonal variations that have been observed by the model previously. This indicates its ability to remember the variations using the LSTM network. These are long-term dependencies which are difficult to capture using simple time-series prediction models. In relation to hourly traffic volume prediction, the RMSE is 16 and the MAPE is 22.29%. In relation to AADT prediction, the RMSE is 42 and the MAPE is 1.36%. Based on the qualitative and quantitative analysis, it can be concluded that the model can indeed capture the long-term variations in the dataset while maintaining model accuracy.

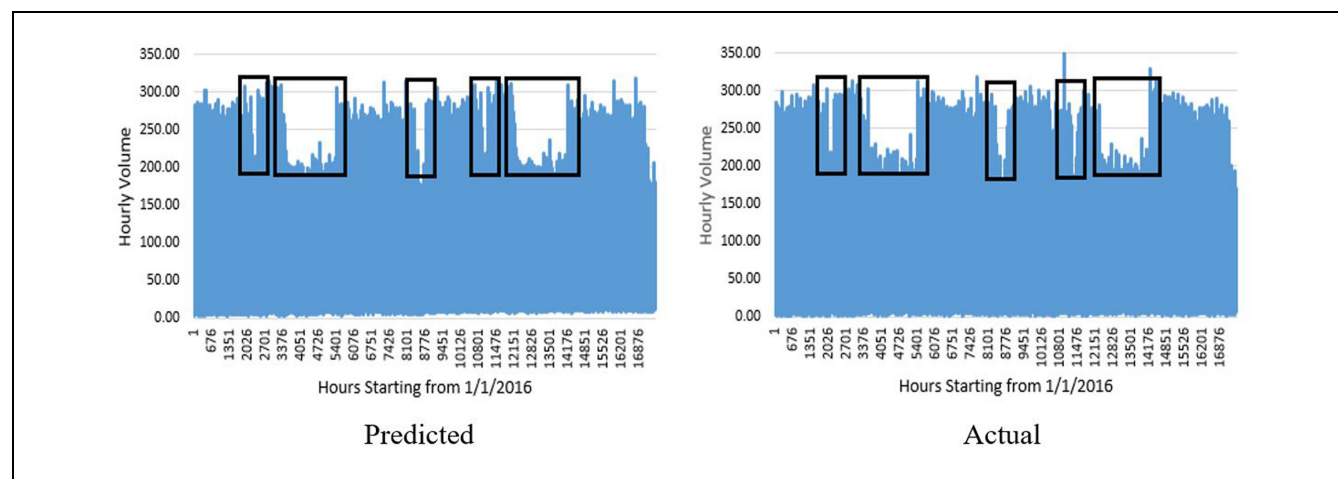


Figure 17. Hourly volume trend (predicted vs. actual) for ATR # 39.

Note: ATR = automatic traffic recorders.

Multi-Year-Ahead Prediction

The LSTM-Median model can be used for AADT prediction in the next year. However, the model can be modified to predict further into the future. A separate experiment was conducted to prove the efficacy of the model for a 2-year-ahead and a 3-year-ahead AADT forecasts. For this step, the LSTM-median model and the ATR #21 were chosen. All the modeling steps remained the same, except the shift operation. The data were shifted by 730 days for a 2-year-ahead forecast, and 1,095 days for a 3-year-ahead forecast. The case of 2-year-ahead forecast is discussed first. The authors are predicting the AADT for 2016 and 2017, but are not able to use the data from 2015 as input when predicting the hourly volumes of 2016. The model loses 1 year of data as a result of the look-ahead scenario. Similarly, for the 3-year-ahead forecast, the model loses 2 years of data.

Table 5 shows the impact of increasing the time horizon of the prediction. For the 1-year-ahead forecast, the LSTM-Median model achieved 99.99% accuracy for 2016 and 98.62% accuracy for 2017. In the 2-year forecast, the accuracy drops to 99.23% and 98.01%, respectively. Finally, for 3-year prediction, the accuracy drops further to 98.04% and 97.78%, respectively. However, accuracies close to 98% are still achieved despite the lack of data from adjacent years. Therefore, this model is effective in predicting AADT into the future.

Contribution of this Research

The major contribution of this research is the development of a novel RNN-based predictive model for high-accuracy AADT and hourly volume prediction. The developed model is capable of capturing the long-term

Table 5. Multi-Year Ahead Forecast with LSTM-Median Model for ATR 21

Model	AADT	Accuracy (%)
Actual (2016)	72,450	
1 year	72,457	99.99
2 years	73,008	99.23
3 years	71,028	98.04
Actual (2017)	72,500	
1 year	73,500	98.62
2 years	73,892	98.01
3 years	70,844	97.78

Note: LSTM = long short-term memory; ATR = automatic traffic recorders; AADT = annual average daily traffic.

variations in the dataset and in tracking seasonal variations. This model was also used to address the issue of missing data using two approaches. A comparative study was performed, and LSTM with median imputation was identified as the best strategy for high-accuracy AADT prediction. To the best of the authors' knowledge, this is the first study involving the development and evaluation of the RNN-based predictive model for future hourly volumes and AADT prediction with missing value treatments. Also, the subsequent investigation of multiple roadway functional groups identifies the best model for each group. Using 92 ATRs, the model was compared with baseline models, achieving the lowest RMSE and MAPE. The hourly traffic volume forecast was accurate and can capture the long-term variations. Finally, results of the multi-year ahead forecast scenario determined that the model maintains a reasonable accuracy for variable time horizons.

Conclusions and Future Study

This study presents the development of a model for AADT prediction that captures the long-term dependencies despite having missing value in the historical dataset. This study also identifies the LSTM-Median model as the best model overall for accurately predicting AADT while capturing the long-term seasonal variations in the time-series. Overall, the LSTM-Median model is deemed the best model in all scenarios for AADT prediction, with an average RMSE of 274 and MAPE of 18.91% for AADT prediction and average RMSE of 824 and MAPE of 2.10% for hourly traffic volume prediction. The model can capture the long-term variations in the dataset while maintaining high accuracy of AADT and hourly traffic volume prediction. The model is also capable of multi-year ahead forecast with trivial reduction in accuracy.

For assessing different RNN models, one ATR was randomly selected from each functional class for the comparison. In future studies, data from a larger sample will be used by including more ATRs from each functional class for evaluating different RNN models. The input data used here are ATR specific. Future work can investigate more complex model architectures based on RNN units that can predict the hourly volumes and AADT for any ATR. This model can be trained on a larger ATR dataset from all ATRs to ensure predictions based on the pattern of the input time-series. Finally, it is possible to expand upon the results of this study in AADT prediction modeling to investigate the use of other RNN models (e.g., Elman RNN, Jordan RNN, and Recurrent Multilayer Perceptron networks) and other imputation methods (e.g., cubic spline interpolation, MA models, and Kalman filters) to develop models with higher accuracy. Moreover, novel deep learning architectures other than RNN can be investigated for hourly volume and AADT prediction. Only ATRs with missing value percentage of less than 20% have been used in this analysis; future models can predict AADT and hourly traffic volume with higher accuracy for all ATRs regardless of the percentage of missing data.

Acknowledgments

The authors acknowledge the South Carolina Department of Transportation, which provided funding for this research.

Author Contributions

The authors confirm the contribution to the paper as follows: study conception and design: ZK, SMK, MC; data collection: ZK, SMK; analysis and interpretation of results: ZK, SMK, MC; draft manuscript preparation: ZK, SMK, MC, KD. All authors reviewed the results and approved the final version of the manuscript.

References

1. Richard, K. R., D. J. Torbic, and D. W. Harwood. Overview of Safety Analyst: Software Tools for Safety Management of Specific Highway Sites. In *Proceedings of the 2007 Mid-Continent Transportation Research Symposium*, Ames, IA, 2007.
2. Lipton, Z. C. *A Critical Review of Recurrent Neural Networks for Sequence Learning*, 2015. <https://arxiv.org/abs/1506.00019>.
3. Li, L., Y. Li, and Z. Li. Missing Traffic Data: Comparison of Imputation Methods. *IET Intelligent Transport Systems*, Vol. 8, No. 1, 2014, pp. 51–57. <https://doi.org/10.1049/iet-its.2013.0052>.
4. Peyre, H., A. Leplège, and J. Coste. Missing Data Methods for Dealing with Missing Items in Quality of Life Questionnaires. A Comparison by Simulation of Personal Mean Score, Full Information Maximum Likelihood, Multiple Imputation, and Hot Deck Techniques Applied to the SF-36 in the French. *Quality of Life Research*, Vol. 20, No. 2, 2011, pp. 287–300. <https://doi.org/10.1007/s11136-010-9740-3>.
5. Che, Z., S. Purushotham, K. Cho, D. Sontag, and Y. Liu. Recurrent Neural Networks for Multivariate Time Series with Missing Values. *Scientific Reports*, Vol. 8, No. 1, 2018, p. 6085. <https://doi.org/10.1038/s41598-018-24271-9>.
6. Anava, O., E. Hazan, and A. Zeevi. Online Time Series Prediction with Missing Data. In *International Conference on Machine Learning*, 2016, pp. 2191–2199.
7. Chatfield, C. *Time-Series Forecasting*. Chapman and Hall/CRC, New York, NY, 2001. <https://doi.org/10.1201/9781420036206>.
8. Långkvist, M., L. Karlsson, and A. Loutfi. A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling. *Pattern Recognition Letters*, Vol. 42, 2014, pp. 11–24. <https://doi.org/10.1016/j.patrec.2014.01.008>.
9. Giles, C. L., S. Lawrence, and A. C. Tsoi. Noisy Time Series Prediction using Recurrent Neural Networks and Grammatical Inference. *Machine Learning*, Vol. 4, No. 1–2, 2001, pp. 161–183. <https://doi.org/10.1023/A:1010884214864>.
10. Han, M., J. Xi, S. Xu, and F. L. Yin. Prediction of Chaotic Time Series Based on the Recurrent Predictor Neural Network. *IEEE Transactions on Signal Processing*, Vol. 52, No. 12, 2004, pp. 3409–3416. <https://doi.org/10.1109/TSP.2004.837418>.
11. Sak, H., A. Senior, and F. Beaufays. *Long Short-Term Memory Recurrent Neural Network Architectures for Large Scale Acoustic Modeling*, 2014. <https://arxiv.org/abs/1402.1128>.
12. Bao, Y., T. Xiong, and Z. Hu. Multi-Step-Ahead Time Series Prediction using Multiple-Output Support Vector Regression. *Neurocomputing*, Vol. 129, 2014, pp. 482–493. <https://doi.org/10.1016/j.neucom.2013.09.010>.
13. Lv, Y., Y. Duan, W. Kang, Z. Li, and F. Y. Wang. Traffic Flow Prediction with Big Data: A Deep Learning Approach. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 16, No. 2, 2014, pp. 865–873. <https://doi.org/10.1109/TITS.2014.2345663>.

14. Huang, W., G. Song, H. Hong, and K. Xie. Deep Architecture for Traffic Flow Prediction: Deep Belief Networks with Multitask Learning. *IEEE Transactions on Intelligent Transportation Systems*, Vol. 15, No. 5, 2014, pp. 2191–2201. <https://doi.org/10.1109/TITS.2014.2311123>.
15. Tian, Y., and L. Pan. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. *Proc., 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, Chengdu, China, IEEE, New York, NY, 2015.
16. Fu, R., Z. Zhang, and L. Li. Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction. In *Proceedings 31: Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, Wuhan, Hubei Province, China, 2016, pp. 324–328.
17. Castro-Neto, M., Y. Jeong, M. K. Jeong, and L. D. Han. AADT Prediction using Support Vector Regression with Data-Dependent Parameters. *Expert Systems with Applications*, Vol. 36, No. 2, 2009, pp. 2979–2986. <https://doi.org/10.1016/j.eswa.2008.01.073>.
18. Šliupas, T. Annual Average Daily Traffic Forecasting using Different Techniques. *Transport*, Vol. 21, No. 1, 2006, pp. 38–43. <https://doi.org/10.1080/16484142.2006.9638039>.
19. Dong, Y., and C. Y. J. Peng. Principled Missing Data Methods for Researchers. *SpringerPlus*, Vol. 2, No. 1, 2013, p. 222.
20. Buuren, S. van, and K. Groothuis-Oudshoorn. Mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*, Vol. 45, No. 3, 2011, pp. 1–68. <https://doi.org/10.18637/jss.v045.i03>.
21. Beretta, L., and A. Santaniello. Nearest Neighbor Imputation Algorithms: A Critical Evaluation. *BMC Medical Informatics and Decision Making*, Vol. 16, Suppl. 3, 2016, p. 74. <https://doi.org/10.1186/s12911-016-0318-z>.
22. Tang, F., and H. Ishwaran. Random Forest Missing Data Algorithms. *Statistical Analysis and Data Mining*, Vol. 10, No. 6, 2017, pp. 363–377. <https://doi.org/10.1002/sam.11348>.
23. Pedregosa, F., R. Weiss, and M. Brucher. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, Vol. 12, 2011, pp. 2825–2830. <https://doi.org/10.1016/j.molcel.2012.08.019>.
24. Asteriou, D., and S. G. Hall. ARIMA Models and the Box–Jenkins Methodology. In *Applied Econometrics* (2nd ed.). Palgrave MacMillan, Basingstoke, UK, 2011.
25. Seabold, S., and J. Perktold. Statsmodels: Econometric and Statistical Modeling with Python. In *Proceedings of the 9th Python in Science Conference*, Vol. 57, 2010, p. 61.
26. Gelper, S., R. Fried, and C. Croux. Robust Forecasting with Exponential and Holt–Winters Smoothing. *Journal of Forecasting*, Vol. 29, No. 3, 2010, pp. 285–300. <https://doi.org/10.1002/for.1125>.
27. Hochreiter, S., and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, Vol. 9, No. 8, 1997, pp. 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>.
28. Dey, R., and F. M. Salem. Gate-Variants of Gated Recurrent Unit (GRU) Neural Networks. *Proc., 2017 IEEE 60th International Midwest Symposium on Circuits and Systems*, Boston, MA, IEEE, New York, NY, 2017.
29. Ketkar, N. Introduction to Keras. In *Deep Learning with Python*. Apress, Berkeley, CA, 2017, pp. 97–111.
30. Abadi, M., P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, X. Zheng, and G. Brain. TensorFlow: A System for Large-Scale Machine Learning. *Proc., 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI '16)*, Savannah, GA, 2016.
31. Kingma, D. P., and J. Ba. *Adam: A Method for Stochastic Optimization*, 2014. <https://arxiv.org/abs/1412.6980>.

The Standing Committee on Highway Traffic Monitoring (ABJ35) peer-reviewed this paper (19-02375).