# Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction

Zhiyong Cui
University of Washington
Seattle, WA 98195, USA
zhiyongc@uw.edu

Ruimin Ke
University of Washington
Seattle, WA 98195, USA
ker27@uw.edu

Yinhai Wang*
University of Washington
Seattle, WA 98195, USA
yinhai@uw.edu

## ABSTRACT

Short-term traffic forecasting based on deep learning methods, especially long-term short memory (LSTM) neural networks, received much attention in recent years. However, the potential of deep learning methods is far from being fully exploited in terms of the depth of the architecture, the spatial scale of the prediction area, and the prediction power of spatial-temporal data. In this paper, a deep stacked bidirectional and unidirectional LSTM (SBU-LSTM) neural network is proposed, which considers both forward and backward dependencies of time series data, to predict the network-wide traffic speed. A bidirectional LSTM (BDLSM) layer is exploited to capture spatial features and bidirectional temporal dependencies from historical data. To the best of our knowledge, this is the first time that BDLSTM is applied as building blocks for a deep architecture model to measure the backward dependency of traffic data for prediction. A comparison with other classical and state-of-the-art models indicates that the proposed SBU-LSTM neural network achieves superior prediction performance for the whole traffic network in both accuracy and robustness.

## KEYWORDS

Deep learning, stacked bidirectional and unidirectional LSTM, network-wide traffic speed prediction

## 1 INTRODUCTION

The performances of intelligent transportation systems (ITS) applications have been largely relying on the quality of traffic information. Recently, with the critical increases in both the total traffic volume and the data they generate, opportunities and challenges coexist in transportation management and research in terms of how to efficiently and accurately understand and exploit the essential information underneath the relatively massive datasets. Short-term traffic forecasting based on data driven models, as a major component in ITS applications, has been one of the most developing research areas in utilizing massive traffic data and has great influence on the overall performance of a variety of modern transportation systems [1].

A large number of methods have been proposed for traffic forecasting in terms of predicting speed, volume, density and travel time since more than three decades ago. Studies in this area normally focus on the methodology part, aiming at developing different models to improve prediction accuracy, efficiency, or robustness. As indicated in previous literature, the existing models can be divided into two categories, i.e. classical statistical methods and computational intelligence (CI) approaches [34]. Most statistical methods for traffic forecasting were proposed at the early stage when traffic condition was less complex and transportation datasets were relatively small in volume. Later on, as the fast development in traffic sensing technologies and computational power, as well as traffic data volume, the majority of the papers developed recently focus on using CI approaches for traffic forecasting.

With the ability to deal with high dimensional data and the capability of depicting non-linear relationship, CI approaches do outperform the statistical methods, such as auto-regressive integrated moving average (ARIMA) [36], with respect to handling complex traffic forecasting problems [38]. However, full potential of artificial intelligence is far from being exploited until the prosperity of the neural networks (NN) based methods. Ever since the precursory study of utilizing NN into the traffic prediction problem was proposed [39], many NN-based methods, like feed forward NN [41], fuzzy NN [40], recurrent NN (RNN) [42], and hybrid NN [25], are adopted for traffic forecasting problems. Due to the dynamic nature of transportation system, RNN, which is adapted to sequence data by maintaining a chain-like structure and the internal memory with loops [4], is especially suitable to capture the temporal evolution of traffic status. The chain-like structure and the depth of the loops make RNNs difficult to train because of the vanishing or blowing up gradient problems during the back-propagating process. There have been a number of attempts to address the difficulty of training RNNs and the vanishing gradients were successfully addressed by the Long Short-Term Memory networks (LSTMs) [3], which is a type of RNN with gated structure to learn long-term dependencies of sequence-based tasks.

As a representative kind of deep learning method handling sequence-data, LSTMs has been gaining popularity in traffic forecasting for its ability to model long-term dependencies. Several studies [2, 22-25, 44, 45] have been done to examine the applicability of LSTMs in traffic forecasting, and the results demonstrate the successfulness and advantage of LSTMs. However, the potential of LSTMs is far from being fully exploited


UrbComp'17, August 14, 2017, Halifax, Nova Scotia, Canada.

in the domain of transportation. This can be summarized from three aspects: 1) the traffic forecasting area has not expanded from a specific location or several adjacent locations to the whole traffic network. 2) Most of the structures of LSTM-based methods are shallow. 3) The long-term dependencies are normally learnt from the input data chronologically arranged, which are forward dependencies, while the backward dependencies learnt from reverse-chronological ordered data has never been explored.

From the perspective of the scale of prediction area, predicting large-scale transportation network traffic has become an important and challenging topic. Most of the existing studies utilize traffic data at a sensor location or along a corridor, and thus, network-wide prediction could not be achieved unless N models were trained for a network with N nodes [22]. To learn the large-scale transportation network traffic, the network structure even can be extracted from the fine-scale grid-segmented images [43]. Thus, learning complex spatial-temporal features of the traffic network by only one model should be explored.

From the point of view of the depth of the structure of LSTM-based models, the structure should have the ability to capture the dynamic nature of the transportation system. Most of the newly proposed LSTM-based prediction models have relatively shallow structures with only one hidden layer to deal with the time series data [2, 22, 44]. Existing studies [20, 21] have shown that deep LSTM architectures with several hidden layers can build up progressively higher level of representations of sequence data. Although some studies [23-25] utilized more than one hidden LSTM layer, the influences of the number of LSTM layers in different LSTM-based models need to be further compared and explained.

In terms of the dependency in prediction problems, the information time series data should be fully utilized. Normally, the datasets fed to the LSTM models are chronologically arranged which results in that the information in the LSTMs is passed in a positive direction from the time step $t-1$ to the time step $t$ along the chain-like structure. Thus, the LSTM structure only makes use of the forward dependencies [5]. But in this process, it is highly possible that useful information is filtered out, or not efficiently passed through the chain-like gated structure. Therefore, it is worth to take the backward dependencies, which passing information in a negative direction, into consideration. Another reason for including backward dependency into our study is the periodicity of traffic. Unlike wind speed forecasting [15], traffic incident forecasting [16], or many other time series forecasting problems with strong randomness, traffic conditions have strong periodicity and regularity, and even short-term periodicity can be observed [17]. Analysing the periodicity of time series data, especially for recurring traffic patterns, from both forward and backward temporal perspectives will enhance the performance [28]. However, based on our review of the literature, few studies on traffic analysis utilized the backward dependency. To fill this gap, a bidirectional LSTM (BDLSTM) with the ability to deal with both forward and backward dependencies is adopted as a component of the network structure in this study.

In addition, when predicting the network-wide traffic speed, other than the speed of a station, the impact of upstream and downstream speeds on each location in the traffic network should not be neglected. Previous studies [26, 27], which only making use of the forward dependencies of time series data, has found that, the past speed values of upstream as well as downstream locations influence the future speed values of a location along a corridor. However, as for complicated traffic networks with intersections and loops, upstream and downstream both refer to relative positions, and two arbitrary locations can be upstream and downstream of each other. Upstream and downstream are with respect to space, while forward and backward dependencies are with respect to time. With the help of forward and backward dependencies of spatial-temporal data, the learnt feature will be more comprehensive.

In this paper, we propose a stacked bidirectional and unidirectional LSTM (SBU-LSTM) neural network, combining LSTM and BDLSTM, for network-wide traffic speed prediction. The selected traffic network consists of four major freeways covering 323 loop detectors stations in Seattle area. Experimental results show that our model achieves network-wide traffic speed prediction with a high prediction accuracy. The influence of the number of layers, the number of time lags and the dimension of weight matrices in LSTM/BDLSTM layers are further analysed. In summary, our contributions can be stated as follows: 1) we expand the traffic forecasting area from a specific location or several adjacent locations along a corridor to the whole traffic network; 2) we adopt the deep architecture and the influence of number of layer is analysed; and 3) backward dependencies are taken into account by combining LSTM and BDLSTM to enhance the feature learning from the large-scale spatial time series data.

## 2 METHODOLOGY

The LSTMs, as a special kind of Recurrent Neural Networks (RNNs), are connectionist models that capture the dynamics of sequence via cycles in the network of nodes [5]. LSTMs have been proved to be able to process sequence data [4] and applied in many real-world problems, like speech recognition [6], image captioning [7], music composition [8] and human trajectory prediction [8]. In this paper, a deep Bidirectional LSTM NN for network-wide traffic speed prediction is proposed and its detailed framework is introduced in this section. Here, speed prediction is defined as predicting future time periods speeds by using historical speeds. The illustrations of the models in the subsections all take the traffic speed prediction as examples.

### 2.1 Network-wide Traffic Speed Data

Traffic speed prediction at one location normally uses a sequence of speed values with $n$ historical time steps as the input data [2, 22, 23], which can be represented by a vector,

$$\boldsymbol{X}_T = [x_{T-n}, x_{T-(n-1)}, \dots, x_{T-2}, x_{T-1}] \quad (1)$$

But the traffic speed at one location may be influenced by the speeds of nearby locations or even locations faraway, especially when traffic jam propagates through the traffic network. To take these network-wide influences into account, the models we propose and compare in this study take the network-wide traffic speed data as the input. Suppose the traffic network consists of $P$ locations and

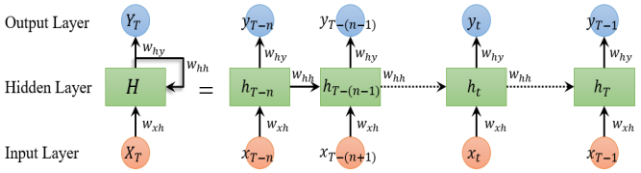


**Figure 1 Standard RNN architecture and an example unfolded in time for T time steps**

we need to predict the traffic speeds at time $T$ using $n$ historical time frames (steps), the input can be characterized as a speed data matrix,

$$\boldsymbol{X}_T^P = \begin{bmatrix} \boldsymbol{x}^1 \\ \boldsymbol{x}^2 \\ \vdots \\ \boldsymbol{x}^P \end{bmatrix} = \begin{bmatrix} x_{T-n}^1 & x_{T-n+1}^1 & \cdots & x_{T-2}^1 & x_{T-1}^1 \\ x_{T-n}^2 & x_{T-n+1}^2 & & x_{T-2}^2 & x_{T-1}^2 \\ & \vdots & \ddots & \vdots & \vdots \\ x_{T-n}^P & x_{T-n+1}^P & \cdots & x_{T-2}^P & x_{T-1}^P \end{bmatrix} \quad (2)$$

where each element $x_t^p$ represents the speed of the $t$-th time frame at the $p$-th location. To reflect the temporal attributes of the speed data and simplify the expressions of the equations in the following subsections, the speed matrix is represented by a vector, $\boldsymbol{X}_T^p = [x_{T-n}, x_{T-(n-1)}, \dots, x_{T-2}, x_{T-1}]$, in which each element is a vector of the $P$ locations' speed values.

## 2.2 RNNs

RNN is a class of powerful deep neural network using its internal memory with loops to deal with sequence data. The architecture of RNNs, which also is the basic structure of LSTMs, is illustrated in Figure 1. For a hidden layer in RNN, it receives the input vector, $\boldsymbol{X}_T^P$, and generates the output vector, $\boldsymbol{Y}_T$. The unfolded structure of RNNs, shown in the right part of Figure 1, presents the calculation process that, at each time iteration, $t$, the hidden layer maintains a hidden state, $h_t$, and updates it based on the layer input, $x_t$, and previous hidden state, $h_{t-1}$, using the following equation:

$$h_t = \sigma_h(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \quad (3)$$

where $W_{xh}$ is the weight matrix from the input layer to the hidden layer, $W_{hh}$ is the weight matrix between two consecutive hidden states ($h_{t-1}$ and $h_t$), $b_h$ is the bias vector of the hidden layer and $\sigma_h$ is the activation function to generate the hidden state. The network output can be characterized as:

$$y_t = \sigma_y(W_{hy}h_t + b_y) \quad (4)$$

where $W_{hy}$ is the weight matrix from the hidden layer to the output layer, $b_y$ is the bias vector of the output layer and $\sigma_y$ is the activation function of the output layer. By applying the Equation (1) and Equation (2), the parameters of the RNN is trained and updated iteratively via the back-propagation (BP) method. In each time step $t$, the hidden layer will generate a value, $y_t$, and the last output, $y_T$, is the desired predicted speed in the next time step, namely $\hat{x}_{T+1} = y_T$.

Although RNNs exhibit the superior capability of modeling nonlinear time series problems [2], regular RNNs suffering from the vanishing or blowing up gradient during the BP process, and thus, being incapable of learning from long time lags [10], or saying long-term dependencies [11].

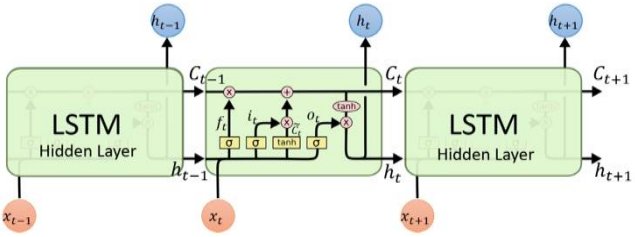


**Figure 2 Hidden layer of LSTM architecture (Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/)**

## 2.3 LSTMs

To handle the aforementioned problems of RNNs, several sophisticated recurrent architectures, like LSTM architecture [3] and Gated Recurrent Unit (GRU) architecture [12] are proposed. It has been showed that the LSTMs work well on sequence-based tasks with long-term dependencies, but GRU, a simplified LSTM architecture, is only recently introduced and used in the context of machine translation [13]. Although there are a variety of typical LSTM variants proposed in recent year, a large-scale analysis of LSTM variant shows that none of the variants can improve upon the standard LSTM architecture significantly [14]. Thus, the standard LSTM architecture is adopted in this study as a part of the proposed network structure and introduced in this section.

The only different component between standard LSTM architecture and RNN architecture is the hidden layer [10]. The hidden layer of LSTM is also named as LSTM cell, which is shown in Figure 2. Like RNNs, at each time iteration, $t$, the LSTM cell has the layer input, $x_t$, and the layer output, $h_t$. The complicated cell also takes the cell input state, $\tilde{C}_t$, the cell output state, $C_t$, and the previous cell output state, $C_{t-1}$, into account while training and updating parameters. Due to the gated structure, LSTM can deal with long-term dependencies to allow useful information pass along the LSTM network. There are three gates in a LSTM cell, including an input gate, a forget gate, and an output gate. The gated structure, especially the forget gate, helps LSTM to be an effective and scalable model for several learning problems related to sequential data [14]. At time $t$, the input gate, the forget gate, and the output gate, denoted as $i_t$, $f_t$, and $o_t$ respectively. The input gate, the forget gate, the output gate and the input cell state, which are represented by yellow boxes in the LSTM cell in Figure 2, can be calculated using the following equations:

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right) \quad (5)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (6)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (7)$$

$$\tilde{C}_t = tanh(W_C x_t + U_o h_{t-1} + b_C) \quad (8)$$

where $W_f$, $W_i$, $W_o$, and $W_C$ are the weight matrices mapping the hidden layer input to the three gates and the input cell state, while the $U_f$, $U_i$, $U_o$, and $U_C$ are the weight matrices connecting the previous cell output state to the three gates and the input cell state.

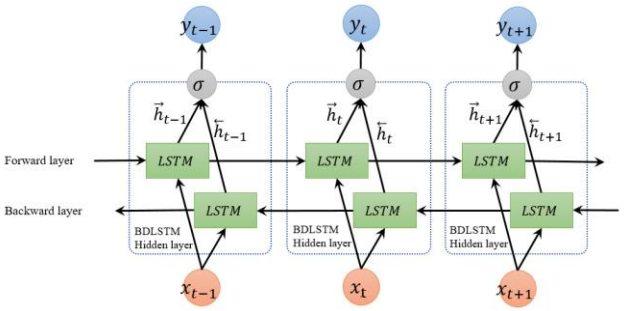


**Figure 3 General architecture of bidirectional LSTM shown unfolded in time for three time-steps**

The $b_f$, $b_i$, $b_o$, and $b_C$ are four bias vectors. The $\sigma_g$ is the gate activation function, which normally is the sigmoid function, and the tanh is the hyperbolic tangent function. Based on the results of four above equations, at each time iteration $t$, the cell output state, $C_t$, and the layer output, $h_t$, can be calculated as follows:

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (9)$$

$$h_t = o_t * \tanh(C_t) \quad (10)$$

The final output of a LSTM layer should be a vector of all the outputs, represented by $\boldsymbol{Y}_T = [h_{T-n}, \dots, h_{T-1}]$. Here, when taking the speed prediction problem as an example, only the last element of the output vector, $h_{T-1}$, is what we want to predict. Thus, the predicted speed value ($\hat{x}$) for the next time iteration, $T$, is $h_{T-1}$, namely $\hat{x}_T = h_{T-1}$.

## 2.4 BDLSTMs

The idea of BDLSTMs comes from bidirectional RNN [18], which processes sequence data in both forward and backward directions with two separate hidden layers. BDLSTMs connect the two hidden layers to the same output layer. It has been proved that the bidirectional networks are substantially better that unidirectional ones in many fields, like phoneme classification [19] and speech recognition [20]. But bidirectional LSTMs have not been used in traffic prediction problem, based on our review of the literature [2,22,23,24,25].

In this section, the structure of an unfolded BDLSTM layer, containing a forward LSTM layer and a backward LSTM layer, is introduced and illustrated in Figure 3. The forward layer output sequence, $\overrightarrow{\boldsymbol{h}}$, is iteratively calculated using inputs in a positive sequence from time $T-n$ to time $T-1$, while the backward layer output sequence, $\overleftarrow{\boldsymbol{h}}$, is calculated using the reversed inputs from time $T-n$ to $T-1$. Both the forward and backward layer outputs are calculated by using the standard LSTM updating equations, Equations (3) - (8). The BDLSTM layer generates an output vector, $\boldsymbol{Y}_T$, in which each element is calculated by using the following equation:

$$y_t = \sigma(\overrightarrow{h}_t, \overleftarrow{h}_t) \quad (11)$$

where $\sigma$ function is used to combine the two output sequences. It can be a concatenating function, a summation function, an average function or a multiplication function. Similar to the LSTM layer, the final output of a BDLSTM layer can be represented by a vector, $\boldsymbol{Y}_T = [y_{T-n}, \dots, y_{T-1}]$, in which the last element, $y_{T-1}$, is the predicted speed value for the next time iteration when taking speed prediction as an example.

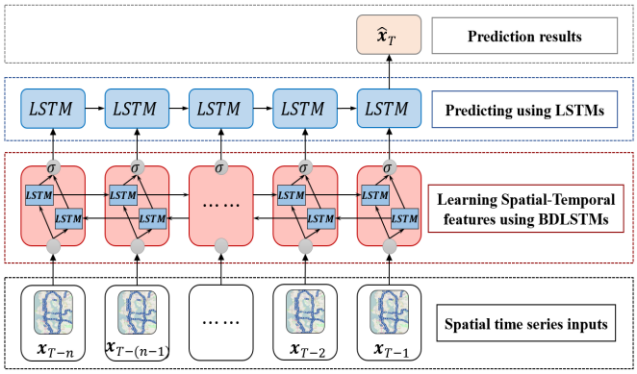


**Figure 4 Architecture of SBU-LSTMs**

## 2.5 Stacked Bidirectional and Unidirectional LSTM Networks

Existing studies [20, 21] have shown that deep (multilayer) LSTM architectures with several hidden layers can build up progressively higher level of representations of sequence data, and thus, work more effective. The deep LSTM architectures are networks with several stacked LSTM hidden layers, in which the output of a LSTM hidden layer will be fed as the input into the next level of LSTM hidden layer. This stacked-layers mechanism, which can enhance the power of neural networks, is adopted in this study. As mentioned in previous sections, BDLSTMs can make use of both forward and backward dependencies. When feeding the spatial-temporal information of the traffic network to the BDLSTMs, both the spatial correlation of the speeds in different locations of the traffic network and the temporal dependencies of the speed values can be captured during the feature learning process. In this regard, the BDLSTMs are very suitable for being the first layer of a model to learn more useful information from spatial time series data. To predict future values, like speed values in this study, the last (highest) layer of a neural network only generates the future (predicted) values. During this process, the last layer only needs to utilize the features and predict future values along the order of the outputs from previous layers. Thus, it is not necessary to adopt the BDLSTM in the last layer.

In this study, we propose a novel deep architecture named stacked bidirectional and unidirectional LSTM network (SBU-LSTM) to predict the network-wide traffic speed values. Figure 4 illustrates the graphical framework of the proposed mode. Each SBU-LSTM contains a BDLSTM layer as the first layer and a LSTM layer as the last layer. For sake of making full use of the input data and learning complex and comprehensive features, the SBU-LSTMs can include one or several optional middle BDLSTM layers, which are not presented in Figure 4, between the first layer and the last layer. Figure 4 shows that the SBU-LSTMs take the spatial time series data as the input and predict future speed values for one time-step. SBU-LSTMs are also capable of predicting

values for multiple future time steps based on historical data. But this property is shown in Figure 4, since the target of this study is to predict network-wide traffic speed for one future time step. The detailed structure of the is described in the experiment section.

# 3 EXPERIMENTS

## 3.1 Dataset Description

In this study, we use the loop detector data, which comes from the Digital Roadway Interactive Visualization and Evaluation Network (DRIVE Net) system [29, 30, 46], as our dataset. The traffic information, which contains speed, volume, and occupancy, is

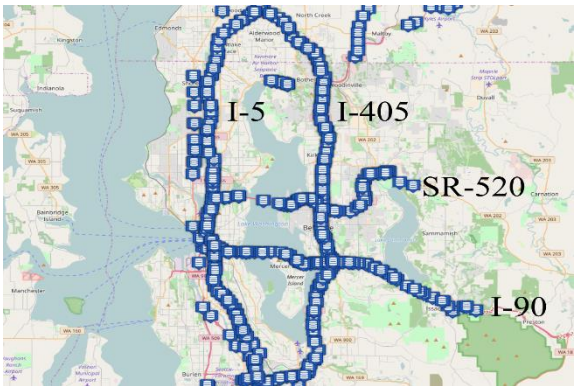


**Figure 5 Loop detector stations on the freeway network in Seattle area**

collected by the single inductive loop vehicle detectors deployed on the freeway system in the Seattle area. The loop detectors are deployed on each lane of the freeway for both directions and connected to a station around every half mile. Due to the replacement and failure of the loop detector stations, a flexible and robust data imputation method is applied to deal with the missing data problem in this dataset [31]. After the data imputation procedure, at each time step, the speed values of all lanes in the same direction at each station are aggregated by taking the average into the station speed value as the basic element of the dataset. The dataset used in this study contains 323 stations extracted from four freeways covering a total of 85 miles, which are I-5, I-405, I-90 and SR-520. in Washington State, shown in Figure 5. Each station is represented by a small blue icon in the figure. Thus, the network-wide traffic is characterized by the 323 station speed values and the spatial dimension of the input data is set as, $P = 323$. The time range of the dataset covers the entire year of 2015, the time interval of the data is 5 minutes, and thus, the dataset has $\frac{60(min)}{5} \times 24(hour) \times 365(year)$ time steps in total.

In the experiments, one unit of the time lag is set as 5 minute. Suppose the number of the time lags is set as $n = 10$, which means the model uses a set of data with 10 consecutive time steps (covering 50 minutes) to predict the future speed value, then the sample size is $N = \frac{60}{5} \times 24 \times 365 - 10 = 105110$. Based on the aforementioned descriptions of the data, each sample of the input data, $\boldsymbol{X}_T^P$, is a 2-D vector with the dimension of $[n, P] = [10, 323]$, and each sample of the output data is a 1-dimension vector with 323 components. Thus, the input of the model is a 3-D vector, whose dimension is $[N, n, P]$. Before fed into the model, all the samples are randomized and divided into training set, validation set, and test set with the ratio 7:2:1.

## 3.2 Experiment Results Analysis and Comparison

In the training process, the model optimizes the mean squared error (MSE) loss using RMSProp optimizer and early stopping mechanism is used to avoid over-fitting. To measure the effectiveness of different travel speed algorithms, the Mean Absolute Errors (MAE) and Mean Absolute Percentage Errors (MAPE) are computed. All the compared models in this section are trained and tested for many times to eliminate outliers, and the results of them presented in this section are averaged to reduce random errors.

In this section, the results of the proposed SBU-LSTMs model are analyzed and compared with classical methods and other RNN-based models. Further analysis about the influence of the number of time lags, the dimension of weight matrices in each layer, and the number of layers are carried out to shed more light on the attributes of LSTM-based models when deal with network-wide traffic speed prediction problems. Since the forget gate is especially important for the LSTM architecture [14], forget gates in the first layer of the proposed model are analyzed in this section. Based on the high prediction accuracy of the proposed model, new applications of this model about traffic pattern analysis are discovered and non-recurring congestions in the traffic network are analyzed.

*1) Comparison with Classical Models for Single Location Traffic Speed Prediction*

Many classical baseline models used in traffic forecasting problems, like ARIMA [2, 23] Support Vector Regression (SVR) [37], Kalman filter [35], are not capable of predicting time series values for 3-D dimension vectors, and thus, they are not able to predict the network-wide traffic speed. To compare with these classical models for traffic prediction, experiment is carried out for each of the loop detector stations, whose input data is a 2-D vector. The results of experiments for single stations are averaged to measure the overall performance of these models. Based on our literature review [2], the performances of ARIMA and Kalman filter method are far behind the others, and thus, these two methods are not compared in this study.

We compared the performance of the SBU-LSTMs with SVR, random forest, feed-forward NN, GRU NN. In this comparison, the proposed model contains only one middle BDLSTM layer. Among these models, the feed-forward NN model, also called Multilayer Perceptron (MLP), has superior performance for the traffic flow prediction [32], and decision tree and SVR are very efficient models for prediction [23, 37]. For the SVR method, the Radial Basis Function (RBF) kernel is utilized, and for the Random Forest method, 10 trees are built and no maximum depth of the trees is limited. The feed-forward NN model has two hidden layers with 323 nodes in each layer.

**Table 1 Prediciton performance of different algorithms for the speed value of only one station. The number of time lag is 10.**

| Models | MAE(mph) | MAPE(%) |
|---|---|---|
| SVM | 9.23 | 20.39 |
| Random Forest | 2.64 | 6.30 |
| Feed-forward NN (2-hidden layers) | 2.63 | 6.41 |
| GRU NN | 3.43 | 8.02 |
| SBU-LSTMs | 2.42 | 5.67 |

Table 1 demonstrates the prediction performance of different algorithms for the single detector station. The number of input time lags in this experiment is set as 10. Among the non-neural network algorithms, random forest performs much better, with the MAE of 2.64, than the SVM method, which makes sense due to the majority votes mechanism of random forest. The feed-forward NN whose MAE is 2.63 performs very close to the random forest method. Although GRU NN is a kind of recurrent NN, its performance obviously cannot outperform those of feed-forward NN and random forest. The single layer and simplified gate structure of GRU NN may be a reason of that. The proposed SBU-LSTMs model is clearly superior to other four methods in this single detector station experiment.

*2) Comparison with LSTM-based models for Network-wide Traffic Speed Prediction*

The SBU-LSTMs is proposed aiming at predicting the network-wide traffic speed, and thus, other methods with the ability of predicting multi-dimensional time series data are compared in this section. Since the proposed model combines BDLSTMs and LSTMs, the deep (N-layers) BDLSTMs and LSTMs are compared. A deep LSTM NN adding a DNN layer, which is proven to be able to boost the LSTM NN [33], is also compared. The DNN layer transforms the output of the LSTM layer to a space that is more discriminative and easier to predict output targets. To measure the influence of temporal information to the network-wide traffic speed, a multilayer LSTM model combining day of week and hour of day is tested in this experiment.

Meanwhile, the influence of depth of the neural networks, namely the number of layers of the models, is tested in this section. All the experiments undertook in this section used the dataset covering the whole traffic network with 10 time lags, which means that 50 minutes' historical data are used to perform forecasting of next 5 minutes. The number of time lags, 10, is set within a reasonable range for traffic forecasting based on literatures [25, 32] and our experiment in the next section. The spatial dimension of weight matrices in each LSTM or BDLSTM layer in this experiment is set as the number of loop detector stations, 323, to ensure the spatial feature can be fully captured. The comparison results are the averaged values of multiple random tests.

Table 2 shows the comparison results, where the headers on horizontal axis show the numbers of the changeable LSTM or BDLSTM layers owned by the models. In terms of the influence of depth of the neural network, all the compared models achieve their best performance when they have two layers and their performances have the same trends that the values of MAE and MAPE increase as the number of layers increases from two to four. In Table 2, the SBU-LSTM is special that it has a 0-layer column customized for it, since the SBU-LSTM originally has two layers. The performance of SBU-LSTM is in conformity with the trends of the compared models that the MAE and MAPE increase as the number of layers rise from zero to four.

The proposed SBU-LSTMs outperform the others for all the layer numbers. When the SBU-LSTM has no middle layer, it achieves the best MAE, 2.426 mph, and MAPE, 5.674%. The test errors of multilayer LSTM NN and BD LSTM NN turn out to be larger than that of the proposed model. They achieve their best MAEs of 2.502 and 2.472, respectively, when the models both have two layers. But for the one-layer case, the BDLSTM NN model gets the worst performance in our experiments shown in the Table 2. It indicates that one-layer BDLSTM may be good enough for capturing features, but it is not satisfactory to predict the results.

**Table 2 Comparison of the proposed architecture over other LSTM-based models for network-wide traffic speed prediction**

| **Model** | **Number of LSTM / BDLSTM layers** | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | N = 0 | | N = 1 | | N = 2 | | N = 3 | | N = 4 | |
| | MAE | MAPE | MAE | MAPE | MAE | MAPE | MAE | MAPE | MAE | MAPE |
| **N**-layers LSTM | | | 2.886 | 6.585 | 2.502 | 5.929 | 2.483 | 5.950 | 2.529 | 6.114 |
| **N**-layers LSTM + 1-layer DNN | | | 2.652 | 6.489 | 2.581 | 6.332 | 2.630 | 6.438 | 2.646 | 6.586 |
| **N**-layers LSTM + Hour of Day + Day of Week | | | 2.668 | 6.506 | 2.557 | 6.274 | 2.595 | 6.447 | 2.647 | 6.602 |
| **N**-layers BDLSTM | | | 3.021 | 6.758 | 2.472 | 5.819 | 2.476 | 5.846 | 2.526 | 5.988 |
| SBU-LSTMs: 1-layer BDLSTM + **N** middle BDLSTM layers + 1-layer LSTM | **2.426** | **5.674** | 2.465 | 5.787 | 2.502 | 5.950 | 2.549 | 6.191 | 2.576 | 6.227 |

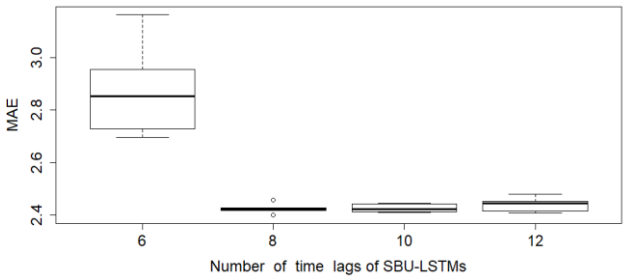


**Figure 6 Boxplot of MAE of different time lags in SBU-LSTMs. One unit of time lag is 5 minutes.**

Except for the one-layer case, the model combining deep LSTM and DNN are not comparable with others. This test result that adding DNN layers to deep LSTM cannot make improvements for the network-wide traffic prediction problem is consistent with the finding in a previous study [33]. The performance of the temporal information added multilayer LSTM NN is very close to, but a little better than, the model combined DNN. It can be seen from the result that incorporating the day of week and time of day features cannot improve the performance. This is in accordance with the results of previous work [23, 24].

*3) Influence of number of time lags and dimension of weight matrices in SBU-LSTMs*

In this study, the dimension of each sample of the input data is $[n, P]$, where $n$ is the temporal dimension demonstrating the number of time lags and $P$ is the spatial dimension representing the number of the loop detector stations. Since the matrix multiplication rule, the spatial dimension of the weight matrices in the first BDLSTM layer of SBU-LSTMs must be accordance with the value of $P$. But the the spatial dimension of weight matrices in other layers can be different and customized. In this section, we measure the influence of the variation of time lags and dimension of weight matrices in SBU-LSTMs with no middle layer.

Figure 6 shows the boxplot of the MAE of different time lags in the proposed SBU-LSTMs, in which the spatial dimensions of all weight matrices are all set as $P$. The MAEs of cases with 8, 10, and12 time lags are very close, around 2.4 and the deviations of the MAEs are relatively small. When the time lag is set as 6, whose period lasts for 30 minutes, the MAE is much higher and the deviation is much larger than other cases, which means 6 time lags is enough not satisfactory for the proposed model. To sum up, for the proposed model and this specific dataset, it is reasonable that more than 30 minutes' historical data should be used to perform forecasting of next 5 minutes.

Table 3 shows the comparison results of SBU-LSTMs with different spatial dimensions of weight matrices in the last LSTM layer. Very close prediction results, MAE and MAPE, are observed from the SBU-LSTM models built with the spatial dimension of $\frac{1}{4}P$, $\frac{1}{2}P$, $P$, $2P$ and $4P$ in the last layer present, where $P$ equals 323. The standard deviations of the multiple test results are nearly the same. That means the variation of the spatial dimension of the weight matrices in the LSTM layer almost has no influence on the prediction results, if the dimension is set as a value close to the number of sensor locations.

**Table 3 Performances comparison of SBU-LSTMs with different spatial dimensions of weight matrices**

| Spatial dimension of weight matrices in the last layer (LSTM layer) | MAE | MAPE | STD |
|---|---|---|---|
| $1/4\ P$ | 2.486 | 5.903 | 0.675 |
| $1/2\ P$ | 2.425 | 5.68 | 0.643 |
| $P = 323$ | 2.426 | 5.674 | 0.63 |
| $2\ P$ | 2.431 | 5.736 | 0.636 |
| $4\ P$ | 2.411 | 5.696 | 0.636 |

*4) Network-wide performance measurement*

Although the MAE and MAPE of the SBU-LSTM outperform those of other methods, the prediction accuracy on each location of the network still need to be further measured. Figure 7 shows two heatmaps of one day's ground truth and predicted speed values of all sensor stations in the traffic network, taking the day of 09/01/2015, a Friday, for an example. As we tested, the average of MAEs and MAPEs of weekday, weekend, and even each day of week are nearly the same. In other words, the performances of SBU-LSTM on predicting network-wide traffic patterns of all days of week are at the same good level. The overall shapes of (dark) red colors areas of both plots in Figure 7 are exactly the same, which means the predicted values of all the locations in the traffic network are very close to the ground truth. Based on the mechanism, at least two aspects of reasons lead the network-wide good performance. One is that the BDLSTM, measuring both forward and backward dependencies, helps learn better features. The other one is that the inherent spatial and temporal correlation between locations is obtained during the training process based on the learnt features.

Slight difference can be observed from the two plots that the heatmap of ground truth data contains more small red points and the heatmap of predicted values seems to be more smooth. The smoothness of predicted values can also be observed in the plots of the ground truth and predicted values of several locations different corridors, shown in Figure 8. It is an advantage of the SBU-LSTMs that the smoothness of the predicted values characterizes more accurate trends of the speed variations in the traffic network. The smoothness of the predicted speed makes the network-wide traffic patterns more visible and intuitive which makes it easier for urban traffic management. Figure 8 shows four plots of comparison ground truth and predictions of four different locations on 09/01/2015. The four subplots in Figure 8 presenting totally different traffic patterns show that the prediction values are pretty accurate, which proves that SBU-LSTM has the ability to predict different patterns over the traffic network at the same time.

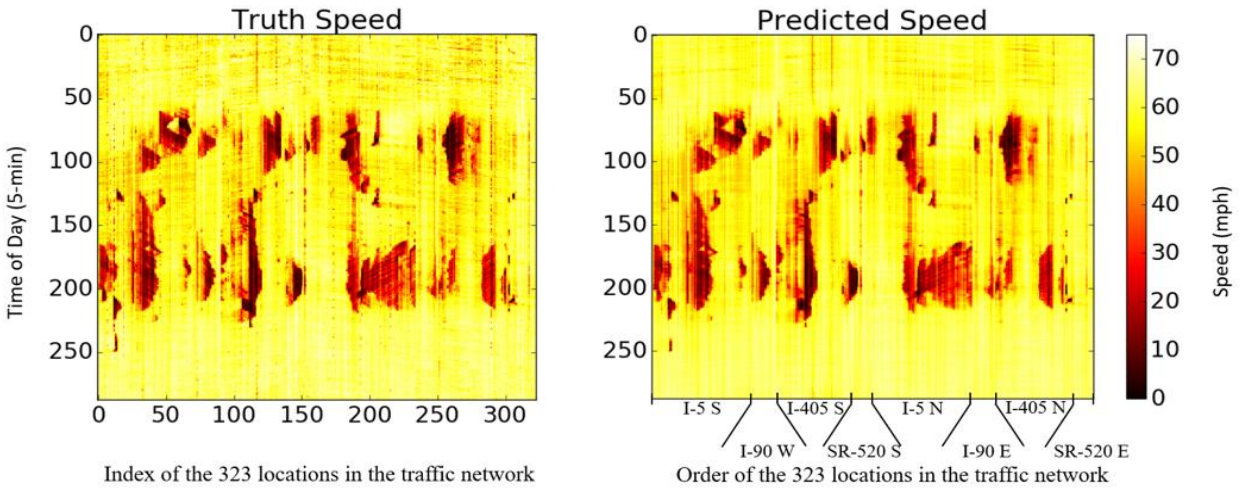


**Figure 7 Heatmaps of ground truth and predicted speed values for the whole network. Taking one day, 01/09/2015, for an example. The two plots share the same meanings of the two axes. The vertical axis represents the time of day in 5-min interval. The horizontal axis represents the index and arrangement order of the 323 sensor stations located on the four corridors, with traffic flow directions presented.**

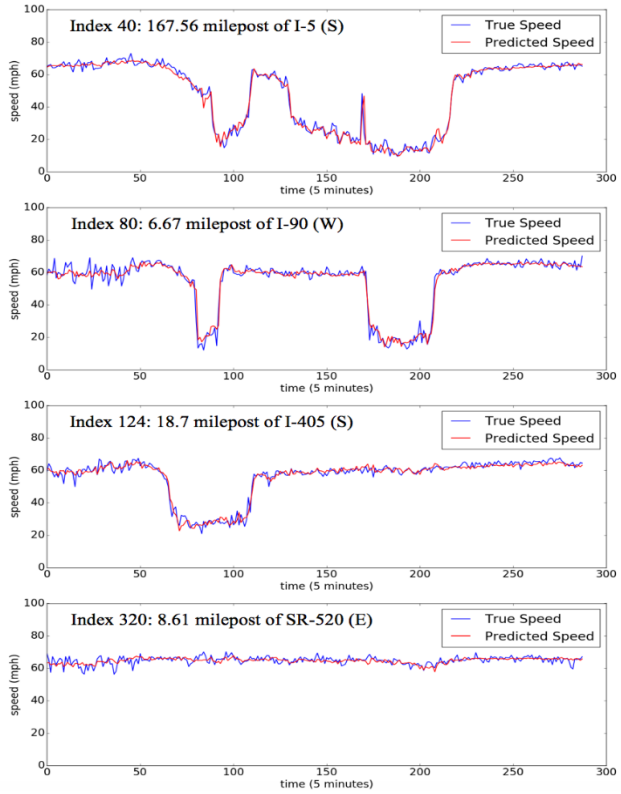


**Figure 8 Comparison between ground truth and predictions of four different locations on 09/10/2015**

## 4 CONCLUSIONS AND FUTURE WORK

A deep stacked bidirectional and unidirectional LSTM neural network is proposed in this paper for network-wide traffic speed prediction. The improvements and contributions in this study mainly focus on three aspects: 1) we expand the traffic forecasting area to the whole traffic network; 2) we adopt the deep stacked architecture and the influence of number of layer is analysed; and 3) both forward and backward dependencies of network-wide traffic data, whose patterns have strong periodicity and regularity, are taken into account.

Experiment results indicate that the two-layers SBU-LSTM without middle layers is the best structure for network-wide traffic speed prediction. Comparing to LSTM, BDLSTM and other LSTM-based methods, the structure of stacking BDLSM and LSTM layers turns out to be more efficient to learn spatial-temporal features from the dataset. If the number of time lags of historical data is not large enough, prediction performance may decrease. But the spatial dimension of weight matrices in the last layer of the model almost has no influence on the prediction results. One advantage of SBU-LSTM is also observed that the curves of predicted speeds are smooth which makes the regularity of traffic pattern more visible and robustness, and thus, makes the urban traffic management more efficient.

Several further improvements and extensions can be carried out based on this study. More complex structures based on SBU-LSTM can be explored and explained. Other factors affecting traffic speed, such as weather, social event and state of roads are deserved to be explored by combining other datasets.

## ACKNOWLEDGMENTS

This work is partially supported by the NSF (Natural Science Foundation) of China under grant number 51138003 and 51329801.

# REFERENCES


[1] Vlahogianni, Eleni I., Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going." Transportation Research Part C: Emerging Technologies 43 (2014): 3-19.

[2] Ma, Xiaolei, et al. "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data." Transportation Research Part C: Emerging Technologies 54 (2015): 187-197.

[3] Hochreiter, Sepp, and Jürgen Schmidhuber. "Long short-term memory." Neural computation 9.8 (1997): 1735-1780.

[4] Zaremba, Wojciech. "An empirical exploration of recurrent network architectures." (2015).

[5] Lipton, Zachary C., John Berkowitz, and Charles Elkan. "A critical review of recurrent neural networks for sequence learning." arXiv preprint arXiv:1506.00019 (2015).

[6] Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton. "Speech recognition with deep recurrent neural networks." Acoustics, speech and signal processing (icassp), 2013 ieee international conference on. IEEE, 2013.

[7] Vinyals, Oriol, et al. "Show and tell: A neural image caption generator." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015.

[8] Eck, Douglas, and Juergen Schmidhuber. "A first look at music composition using lstm recurrent neural networks." Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale 103 (2002).

[9] Alahi, Alexandre, et al. "Social lstm: Human trajectory prediction in crowded spaces." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[10] Gers, Felix A., Jürgen Schmidhuber, and Fred Cummins. "Learning to forget: Continual prediction with LSTM." Neural computation 12.10 (2000): 2451-2471.

[11] Bengio, Yoshua, Patrice Simard, and Paolo Frasconi. "Learning long-term dependencies with gradient descent is difficult." IEEE transactions on neural networks 5.2 (1994): 157-166.

[12] Cho, Kyunghyun, et al. "On the properties of neural machine translation: Encoder-decoder approaches." arXiv preprint arXiv:1409.1259 (2014).

[13] Chung, Junyoung, et al. "Empirical evaluation of gated recurrent neural networks on sequence modeling." arXiv preprint arXiv:1412.3555 (2014)

[14] Greff, Klaus, et al. "LSTM: A search space odyssey." IEEE transactions on neural networks and learning systems (2016).

[15] Wang, Meng-Di, Qi-Rong Qiu, and Bing-Wei Cui. "Short-term wind speed forecasting combined time series method and arch model." Machine Learning and Cybernetics (ICMLC), 2012 International Conference on. Vol. 3. IEEE, 2012.

[16] Zheng, Xiaoping, and Mengting Liu. "An overview of accident forecasting methodologies." Journal of Loss Prevention in the process Industries 22.4 (2009): 484-491.

[17] Jiang, Xiaomo, and Hojjat Adeli. "Wavelet Packet-Autocorrelation Function Method for Traffic Flow Pattern Analysis." Computer-Aided Civil and Infrastructure Engineering 19.5 (2004): 324-337.

[18] Schuster, Mike, and Kuldip K. Paliwal. "Bidirectional recurrent neural networks." IEEE Transactions on Signal Processing 45.11 (1997): 2673-2681.

[19] Graves, Alex, and Jürgen Schmidhuber. "Framewise phoneme classification with bidirectional LSTM and other neural network architectures." Neural Networks 18.5 (2005): 602-610.

[20] Graves, Alex, Navdeep Jaitly, and Abdel-rahman Mohamed. "Hybrid speech recognition with deep bidirectional LSTM." Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on. IEEE, 2013.

[21] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." Nature 521.7553 (2015): 436-444.

[22] Duan, Yanjie, Yisheng Lv, and Fei-Yue Wang. "Travel time prediction with LSTM neural network." Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, 2016.

[23] Chen, Yuan-yuan, et al. "Long short-term memory model for traffic congestion prediction with online open data." Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on. IEEE, 2016.

[24] Wu, Yuankai, and Huachun Tan. "Short-term traffic flow forecasting with spatial-temporal correlation in a hybrid deep learning framework." arXiv preprint arXiv:1612.01022 (2016).

[25] Yu, Rose, et al. "Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting."

[26] Chandra, Srinivasa Ravi, and Haitham Al-Deek. "Predictions of freeway traffic speeds and volumes using vector autoregressive models." Journal of Intelligent Transportation Systems 13.2 (2009): 53-72.

[27] Kamarianakis, Yiannis, H. Oliver Gao, and Poulicos Prastacos. "Characterizing regimes in daily cycles of urban traffic using smooth-transition regressions." Transportation Research Part C: Emerging Technologies 18.5 (2010): 821-840.

[28] Box, George EP, et al. Time series analysis: forecasting and control. John Wiley & Sons, 2015.

[29] Cui, Zhiyong, et al. "New progress of DRIVE Net: An E-science transportation platform for data sharing, visualization, modeling, and analysis." Smart Cities Conference (ISC2), 2016 IEEE International. IEEE, 2016.

[30] Ma, Xiaolei, Yao-Jan Wu, and Yinhai Wang. "DRIVE Net: E-science transportation platform for data sharing, visualization, modeling, and analysis." Transportation Research Record: Journal of the Transportation Research Board 2215 (2011): 37-49.

[31] Henrickson, Kristian, Yajie Zou, and Yinhai Wang. "Flexible and robust method for missing loop detector data imputation." Transportation Research Record: Journal of the Transportation Research Board 2527 (2015): 29-36.

[32] Lv, Yisheng, et al. "Traffic flow prediction with big data: a deep learning approach." IEEE Transactions on Intelligent Transportation Systems 16.2 (2015): 865-873.

[33] Sainath, Tara N., et al. "Convolutional, long short-term memory, fully connected deep neural networks." Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on. IEEE, 2015.

[34] Vlahogianni, Eleni I., Matthew G. Karlaftis, and John C. Golias. "Short-term traffic forecasting: Where we are and where we're going." Transportation Research Part C: Emerging Technologies 43 (2014): 3-19.

[35] Guo, Jianhua, Wei Huang, and Billy M. Williams. "Adaptive Kalman filter approach for stochastic short-term traffic flow rate prediction and uncertainty quantification." Transportation Research Part C: Emerging Technologies 43 (2014): 50-64.

[36] Ye, Qing, Wai Yuen Szeto, and Sze Chun Wong. "Short-term traffic speed forecasting based on data recorded at irregular intervals." IEEE Transactions on Intelligent Transportation Systems 13.4 (2012): 1727-1737.

[37] Wu, Chun-Hsin, Jan-Ming Ho, and Der-Tsai Lee. "Travel-time prediction with support vector regression." IEEE transactions on intelligent transportation systems 5.4 (2004): 276-281.

[38] Karlaftis, Matthew G., and Eleni I. Vlahogianni. "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights." Transportation Research Part C: Emerging Technologies 19.3 (2011): 387-399.

[39] Hua, Jiuyi, and Ardeshir Faghri. "AppHcations of Artificial Neural Networks to Intelligent Vehicle-Highway Systems." RECORD 1453 (1994): 83.

[40] Yin, Hongbin, et al. "Urban traffic flow prediction using a fuzzy-neural approach." Transportation Research Part C: Emerging Technologies 10.2 (2002): 85-98.

[41] Park, Dongjoo, and Laurence R. Rilett. "Forecasting freeway link travel times with a multilayer feedforward neural network." Computer-Aided Civil and Infrastructure Engineering 14.5 (1999): 357-367.

[42] Van Lint, J., S. Hoogendoorn, and H. Van Zuylen. "Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks." Transportation Research Record: Journal of the Transportation Research Board 1811 (2002): 30-39.

[43] Yu, Haiyang, et al. "Spatiotemporal Recurrent Convolutional Networks for Traffic Prediction in Transportation Networks." arXiv preprint arXiv:1705.02699 (2017).

[44] Fu, Rui, Zuo Zhang, and Li Li. "Using LSTM and GRU neural network methods for traffic flow prediction." Chinese Association of Automation (YAC), Youth Academic Annual Conference of. IEEE, 2016.

[45] Zhao, Zheng, et al. "LSTM network: a deep learning approach for short-term traffic forecast." IET Intelligent Transport Systems 11.2 (2017): 68-75.

[46] Wang, Yinhai, et al. Digital Roadway Interactive Visualization and Evaluation Network Applications to WSDOT Operational Data Usage. Diss. University of Washington Seattle, Washington, 2016.