2019

Applications of Deep Learning Models for Traffic Prediction Problems

Rezaur Rahman
University of Central Florida

APPLICATIONS OF DEEP LEARNING MODELS FOR TRAFFIC PREDICTION PROBLEMS

by

REZAUR RAHMAN
B.Sc. Bangladesh University of Engineering and Technology, 2015

A thesis submitted in partial fulfillment of the requirements
for the degree of Master of Science
in the Department of Civil, Environmental and Construction Engineering
in the College of Engineering and Computer Science
at the University of Central Florida
Orlando, Florida

Spring Term
2019

Major Professor: Samiul Hasan

ABSTRACT

Deep learning coupled with existing sensors based multiresolution traffic data and future connected technologies has immense potential to improve traffic operation and management. But to deal with complex transportation problems, we need efficient modeling frameworks for deep learning models. In this study, we propose two different modeling frameworks using Deep Long Short-Term Memory Neural Network (LSTM NN) model to predict future traffic state (speed and signal queue length).

In our first problem, we present a modeling framework using deep LSTM NN model to predict traffic speeds in freeways during regular traffic condition as well as under extreme traffic demand, such as a hurricane evacuation. The approach is tested using real-world traffic data collected during hurricane Irma's evacuation for the interstate 75 (I-75), a major evacuation route in Florida. We perform several experiments for predicting speeds for 5 min, 10 min, and 15 min ahead of current time. The results are compared against other traditional prediction models such as *K-Nearest Neighbor*, Analytic Neural Network (ANN), Auto-Regressive Integrated Moving Average (ARIMA). We find that LSTM-NN performs better than these parametric and non-parametric models. Apart from the improvement in traffic operation, the proposed method can be integrated with evacuation traffic management systems for a better evacuation operation.

In our second problem, we develop a data-driven real-time queue length prediction technique using deep LSTM NN model. We consider a connected corridor where information from vehicle detectors (located at the intersection) will be shared to consecutive intersections. We assume that the queue length of an intersection in the next cycle will depend on the queue length of the target and two upstream intersections in the current cycle. We use InSync Adaptive Traffic Control

System (ATCS) data to train a Long Short-Term Memory Neural Network model capturing time-dependent patterns of a queue of a signal. To select the best combination of hyperparameters, we use sequential model-based optimization (SMBO) technique. Our experiment results show that the proposed modeling framework performs very well to predict the queue length. Although we run our experiments predicting the queue length for a single movement, the proposed method can be applied for other movements as well. Queue length prediction is a crucial part of an ATCS to optimize control parameters and this method can improve the existing signal optimization technique for ATCS.

Keywords: Deep-learning, Long short-term memory, Data-driven, Traffic state, Real-time queue length, Adaptive Traffic Control System.

ACKNOWLEDGMENT

I would like to convey my heartiest gratitude to my honorable supervisor Dr. Samiul Hasan for his excellent supervision and constant support in this thesis. I would also like to thank Dr. Mohamed Abdel-Aty and Yaobang Gong for their help in collecting data from different sources. Finally, I would like to acknowledge the support and encouragement from my family and friends.

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 Introduction

Traffic congestion is a serious problem in most of the urban areas. In 2011, it caused urban Americans to spend 6.9 billion hours more in traveling and cost them an extra 3.1 billion gallons of fuel, for a congestion cost of $160 billion (Schrank. et al., 2015). Inability to estimate future traffic state for proactive decision making, inefficiencies in traffic management and control, for example, poor inaccurate queue estimation for traffic signal timing, and lack of coordination between adjacent intersections are a few major causes of congestion problem (Smith et al., 2013). Ability to provide accurate information about current and future traffic state will help to overcome these challenges. Moreover, accurate traffic state prediction can enhance traffic management systems (TMS) by giving opportunities to the transportation agencies to react proactively to overcome recurrent and non-recurrent congestion and changes in traffic conditions.

Predicting traffic states in real-time needs traffic data from various sources. Many transportation agencies have deployed various traffic sensors such as Loop Detectors, Bluetooth, Magnetic Vehicle Detection System (MVDS), Video-Based detection, etc. in their transportation systems. These sensors allow us to collect multi-resolution traffic data in real-time and recognize patterns for estimating traffic states.

Moreover, in recent years, advances in wireless communication systems have created a new horizon in traffic operations and management. Advanced wireless communication technologies such as Wi-fi, WiMAX, LTE, and DSRC create an opportunity to develop a connected environment where vehicles are connected with each other (V2V) and with the infrastructures (V2I). This system will generate a large amount of data regarding traffic states, vehicle positions, delays, etc.

Therefore, the future of transportation will largely depend on data-driven solution for different problems such as traffic state prediction for highways and arterials, data-driven performance measures and control parameters optimization for signal timing, etc. But to deal with these problems, we need reliable models that can capture traffic flow patterns with better accuracy.

Recent trends in transportation research show that researchers are exploring sensor-based data-driven approaches to solve different transportation-related problems since these approaches are easy to deploy in a real-time context. A few commonly used data-driven approaches include support vector machine (SVM), *k*-nearest neighbor (KNN), analytic neural network (ANN), ARIMA, etc. These models perform reasonably well for predicting traffic states (speed, travel time, traffic flow, etc.) (Billings and Jiann-Shiou, 2006; Deshpande and Bajaj, 2016; Lee, 2009; C. H. Wu et al., 2004; Yu et al., 2016).

Deep learning is one of the most recent innovations in machine learning. It can capture the sharp discontinuities in traffic flows using multilayered non-linear functions (tanh, sigmoid etc.) (Polson and Sokolov, 2017). Applications of deep learning models in transportation will allow us to deal with more complex problems and big data (Rahman and Hasan, 2019, 2018).

1.2 Thesis Contribution

This study has made several contributions towards traffic operation and management by improving the existing short-term traffic prediction methods. It also investigates the irregular pattern in traffic flow behavior in an extreme traffic demand condition such as hurricane evacuation. Unlike the existing time series prediction problem, we develop a modeling framework to capture the complex dynamics in traffic flow considering both spatial and temporal dependency of the traffic flow behavior. This method can predict the traffic speed at different time horizon with better accuracy, which can largely improve traffic management, especially during evacuation by allowing proactive decision making.

Another part of the thesis presents a new approach for real-time signal queue length prediction considering future connectivity (V2V and V2I communication). We develop a data-driven method using deep LSTN NN model for signal queue length prediction. This method will reduce the dependency of the ATCS on multiple detectors (loop detectors, video camera-based detection, etc.) for queue length estimation, hence reducing the overall maintenance cost to operate a system. The approach has been tested using inSync adaptive signal data and can also be used to develop data-driven optimization technique for adaptive traffic control.

1.3 Objectives of the Thesis

The focus of this study is to evaluate the performance of deep learning model while dealing with complex traffic operation problems. We consider two different problems related to traffic state prediction. The main objectives of this study are:

- To develop a framework to predict the traffic state (speed, queue length) considering spatial and temporal dependency of the traffic pattern
- To evaluate the performance of deep learning model in traffic prediction and compare it with traditional machine learning models.
- To check the reliability of the model in heavy demand condition such as hurricane evacuation.

1.4 Thesis Organization

The rest of the thesis is organized as follows: Chapter 2 provides a brief discussion on deep neural networks models. Chapter 3 provides the data description, analysis, methodology and result for short term traffic speed prediction. Chapter 4 describes the data description, methodology, and result for traffic signal queue length prediction. Chapter 5 presents the summary and conclusions of the thesis.

CHAPTER 2: DEEP NEURAL NETWORK MODELS

2.1 Introduction

Deep-learning is a part of broader family of machine learning methods. The basic difference is between deep learning and machine learning is that machine learning methods are task-specific while deep learning methods are based on learning data representations (Lecun et al., 2015). Deep learning methods consist of non-linear modules that transform the raw data representation at one level (starting with the raw input) into representation at a higher, slightly more abstract level. Which allows very complex functions to be learned. Therefore, Deep learning has created a unique opportunity to deal with more complex problems. Deep learning is a recent innovation in machine learning research which emerged as a powerful tool due to a tremendous increase in computational power and data availability. In this chapter, we briefly discussed three different deep learning models.

2.2 Feed Forward Neural Network

The core architecture for modern deep learning methods is based on classical artificial neural networks (ANNs). Though the design of ANNs was inspired by the structure of a real brain, the processing elements and the architecture used in ANN have gone far from their biological inspiration (Svozil et al., 1997). ANNs are versatile, powerful, and scalable which makes them ideal to tackle large and highly complex machine learning tasks.

Feed Forward Neural Network (FFNN) composed of one input layer, one or more hidden layers and one final output layer (Figure 2.2). The hidden and output layers consist of linear threshold units. Every layer except the output layer includes a bias neuron and is fully connected to the next layer. When an ANN has two or more hidden layers, it is called a deep neural network (DNN).

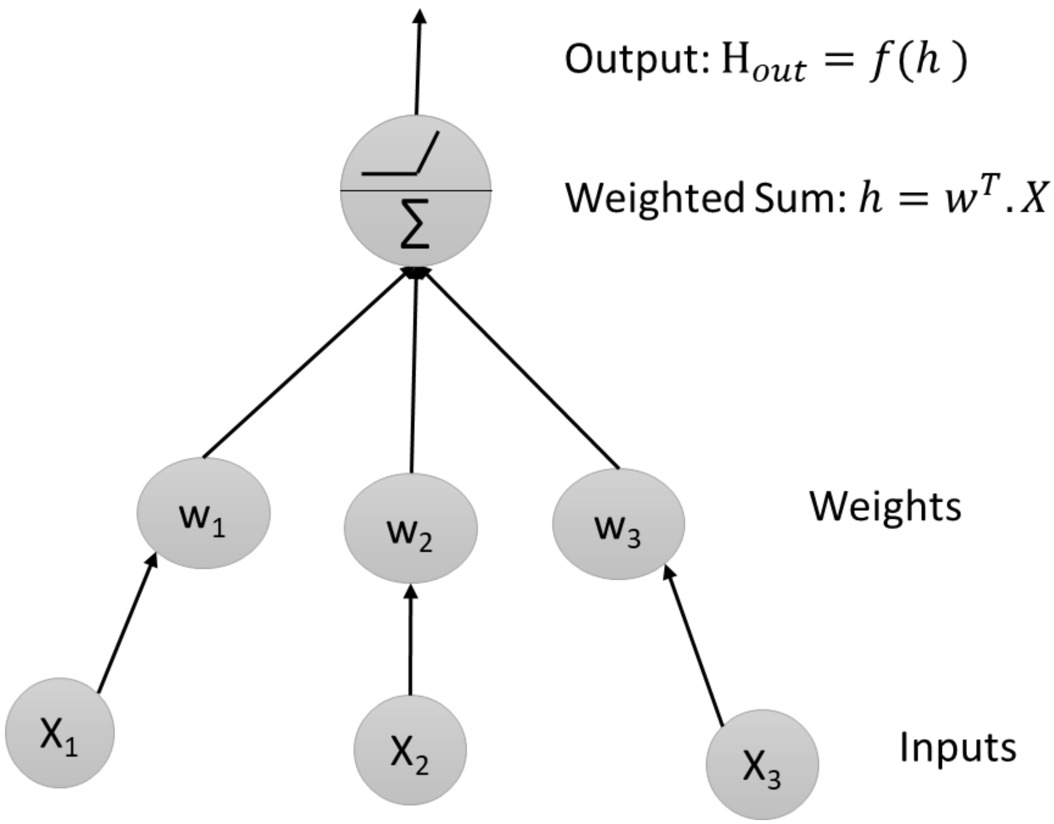
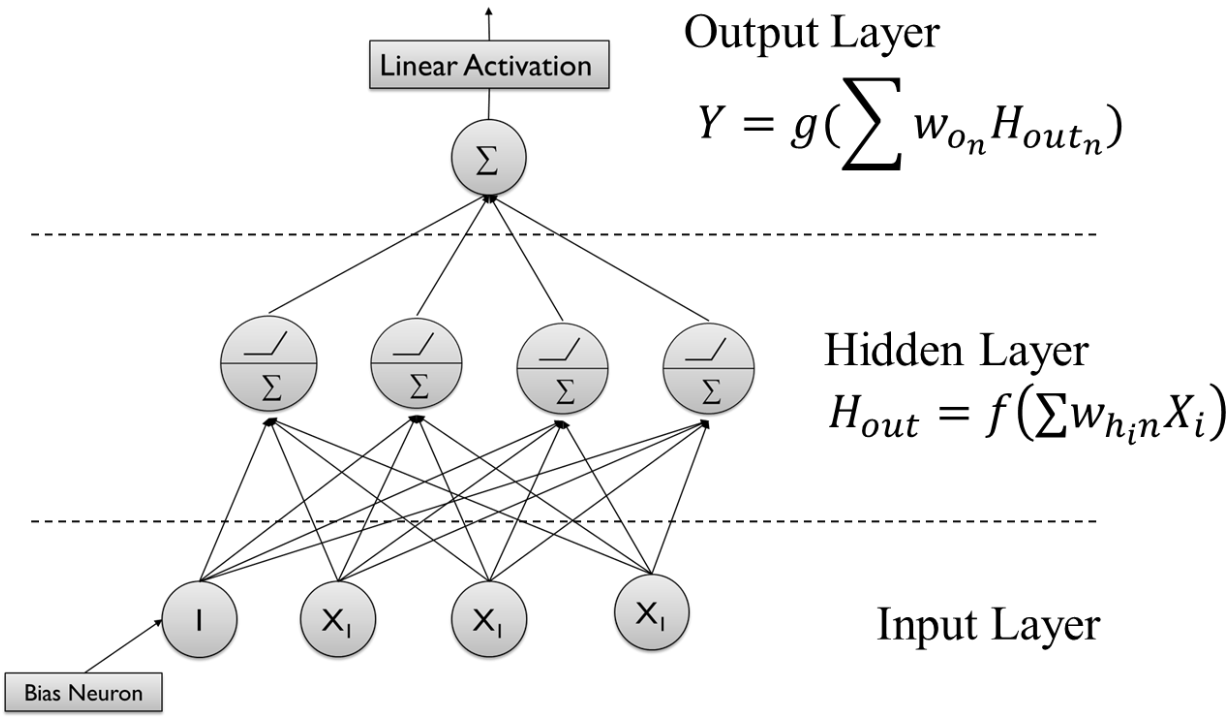


Figure 2.1: Single Hidden Neuron with Nonlinear Activation Function



f = *hidden layer activation function*
w_{h_i} = *weight for the hidden nodes*
g = *ouput layer activationfunction*(*linear or softmax*)
w_{o_n} = weights from output node

Figure 2.2: Feed Forward Neural Network Structure for Prediction

Each training instance of a FFNN can be divided into three steps, forward pass, reverse pass and gradient descent (Geron, 2017). In forward pass step, the backpropagation algorithm makes a prediction and measures the output error (difference between desire and actual output) how much each neuron in the last hidden layer contributed to each output neuron's error. In reverse pass step, it goes through each layer in reverse direction to measure the error contribution from each neuron in the previous hidden layers until the algorithm reaches the input layer. In the gradient descent step, the backpropagation algorithm readjusts the connection weights to reduce the error. The key change in the classical architecture was changing the step function with logistic activation function (f), $\frac{1}{1+exp(-z)}$ (Figure 2.1). Logistic function has a well-defined nonzero derivative which allows gradient descent to make progress during propagation over the layers. However, the backpropagation algorithm can be implemented using other activation functions such as hyperbolic tangent, Rectified linear Unit (ReLU) etc.

2.3 Recurrent Neural Network

The basic concept of Recurrent Neural Network (RNN) is that it stores relevant parts of the input variables and use this information to predict output in the future. RNNs repetitively perform the same computational operation on every element of a sequence and each output is calculated based on the previous computations (Figure 2.2). An RNN can process sequential data very well (Xu et al., 2017).

As shown in Figure 2.2, an RNN can be considered as a chain of repeating modules. In standard RNNs, this repeating module will have a very simple structure, such as a single tanh layer. Hidden state or memory cell of this structure preserves information from the previous input variables. At time step t, the memory cell's current state (h_t) is a function of input state vectors at that current time step (X_t) and hidden state at the previous time step (h_{t-1}), so $\mathrm{h(t)} = \mathrm{f}(h_{t-1}, X_t)$. Its output at time step t, denoted by (y_t), is also a function of the previous state and the current input (Figure 2.2). For basic cells, the output (y_t) and the hidden state (h_t) at a given time step are same.

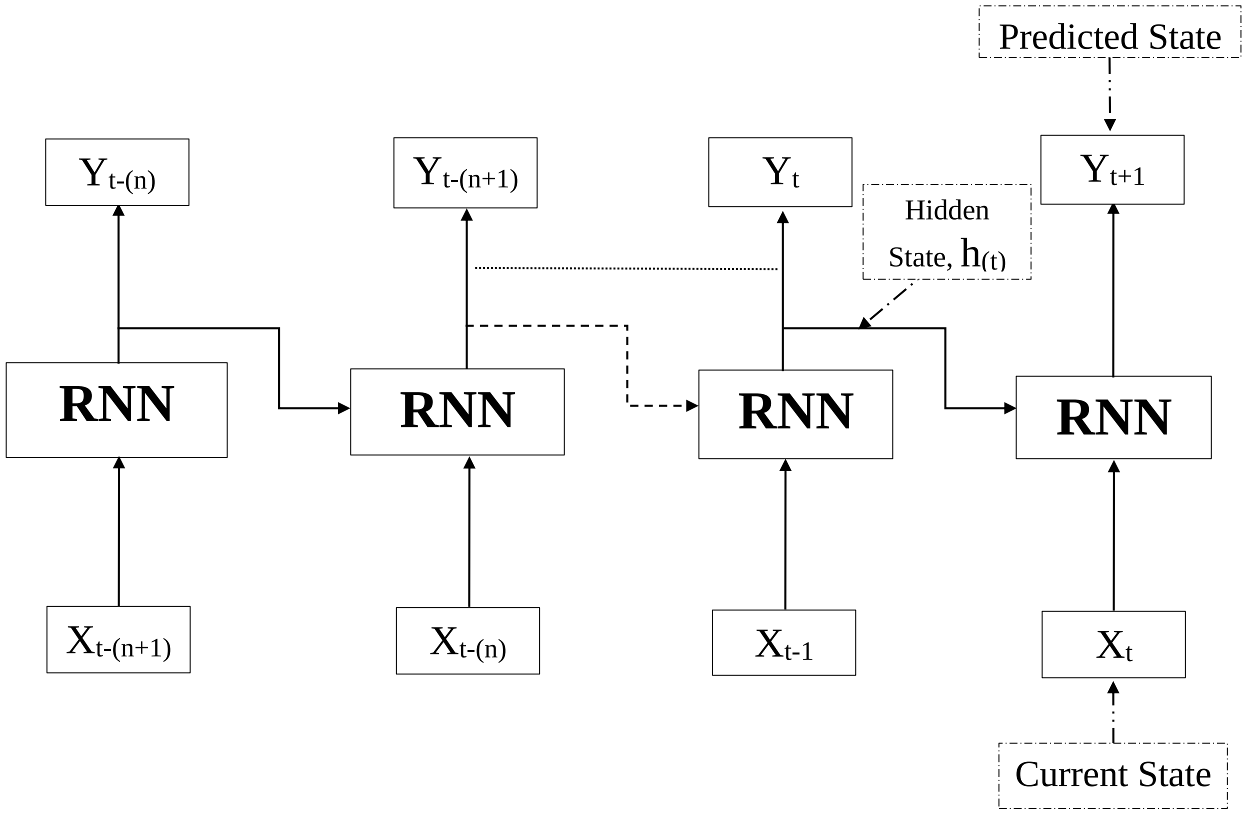


Figure 2.3: A Recurrent Neuron Network Unrolled through Time

Although RNNs can better capture nonlinearity in time series problems, they are weak on learning long-term dependencies due to vanishing of gradient during the backpropagation process (Gers and Cummins, 1999, Hochreiter and Urgen Schmidhuber, 1997). Moreover, traditional RNNs learn a time series sequence based on a predetermined time lag, but it is difficult to find an optimal time window size in an automatic way (Gers and Cummins, 1999), Ma et al., 2015).

2.4 Long Short-Term Memory Neural Networks

To overcome the disadvantages of RNNs, Hochreiter and Schmidhuber proposed the architecture of Long Short-Term Memory Neural Network (LSTM-NN) and an appropriate gradient-based algorithm to solve it (Hochreiter and Urgen Schmidhuber, 1997). The primary objectives of

LSTM-NN are to capture long-term dependencies and determine the optimal time lag for time series problems.

In an LSTM, the cell state (hidden State) is divided into two states: short-term state (h_t) (similar to an RNN) and long-term state (c_t). The long-term state (c_t) stores the information to capture the long-term dependencies among current hidden state and previous hidden states over time. Traversing from the left to the right, the long-term state passes through a forget gate and drops some memories and then adds some new memories via an addition operation (Figure 2.4 and 2.5).

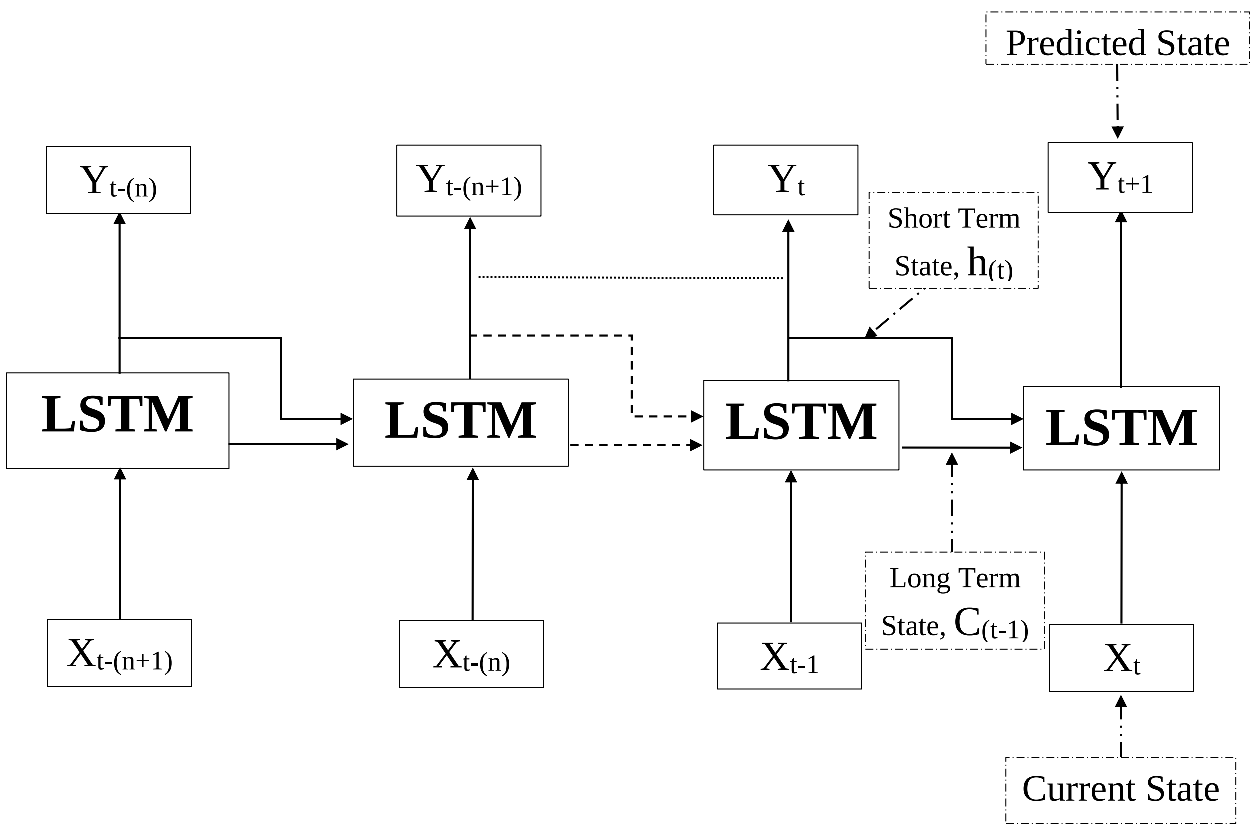


Figure 2.4: Long Short-Term Memory Neural Network Unrolled Over Time

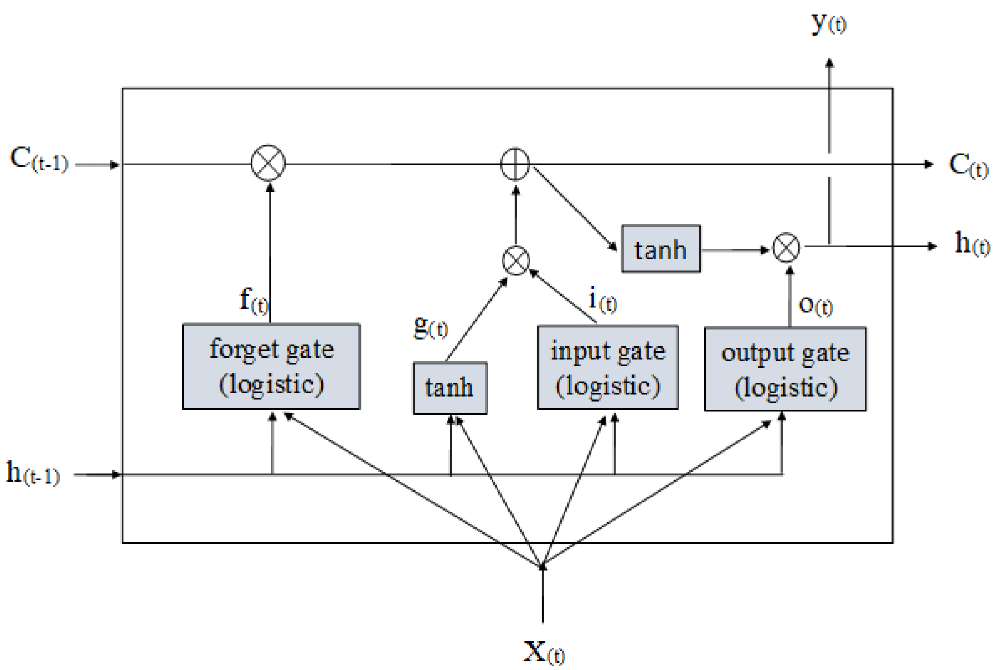


Figure 2.5: Complete Structure of LSTM Cell

As shown in Figure 2.5, a fully connected LSTM cell contains four layers (sigma and tanh) and the input vector (X_t) and the previous short-term state (h_{t-1}) are fed into these layers. The main layer uses tanh activation functions which outputs $(g_{(t)})$. The output from this layer is partially stored in long-the term state $(c_{(t)})$. The other three layers are gate controller user logistic activation function and their output ranges from 0 to 1. The forget state $f_{(t)}$ control which parts of the long-term state should be erased while input gate $i_{(t)}$ decide which parts of the input should be added. The output gate o(t), finally controls which parts of the long-term state should be read and output at this time step $y_{(t)}$ ($=h_{(t)}$). The equations for these operations can be written as follows,

Input gate:

$$\mathrm{i(t)} = \sigma(\mathrm{W}_{xi}^{T}.\mathrm{x(t)} + \mathrm{W}_{hi}^{T}.h(t-1) + b_i) \tag{1}$$

Forget gate:

$$\mathrm{f(t)} = \sigma(\mathrm{W}_{xf}^{T}.\,\mathrm{x(t)} + \mathrm{W}_{hf}^{T}.\,h(t-1) + b_f) \tag{2}$$

Output gate:

$$\mathrm{o(t)} = \sigma(\mathrm{W}_{xo}^{T}.\,\mathrm{x(t)} + \mathrm{W}_{ho}^{T}.\,h(t-1) + b_o) \tag{3}$$

Cell input:

$$\mathrm{g(t)} = \tanh(\mathrm{W}_{xg}^{T}.\,\mathrm{x(t)} + \mathrm{W}_{hg}^{T}.\,h(t-1) + b_g) \tag{4}$$

Where, W_{xi} , W_{xf}, W_{xo}, W_{xg} are the weight matrices of the each of the four layers for their connection to the input vector $X_{t,}$ W_{hi} , W_{hf}, W_{ho}, W_{hg} are the weight matrices of the each of the four layers for their connection to the short-term state (h_{t-1}) and b_i, b_f, b_o, b_c are the bias terms for each of the four layers, σ represents the sigmoid function $\frac{1}{1+exp(-x)}$ and $\tanh$ represents the hyperbolic tangent function $\frac{exp(x)-exp(-x)}{exp(x)+exp(-x)}$. Finally, the long-term and short-term state are calculated using following equations,

Long-term state:

$$c_{(t)} = f_{(t)} \otimes c_{(t-1)} + i_{(t)} \otimes g_{(t)} \tag{5}$$

Short-term state:

$$y_{(t)} = h_{(t)} = \mathrm{o(t)} \otimes tanh(c_{(t)}) \tag{6}$$

CHAPTER 3: SHORT TERM TRAFFIC SPEED PREDICTION FOR FREEWAYS

3.1 Introduction and motivation

Short term traffic state prediction concerns the prediction of traffic state from a few seconds to possibly a few hours into the future (Vlahogianni et al., 2014a). As an integral part, most of the intelligent transportation systems short term traffic state prediction is very crucial in traffic operation for proactive decision making. Especially in a scenario when we have to deal with a heavy traffic demand such as hurricane evacuation. Hurricane causes severe traffic disruption and loss in human mobility (Roy, 2018; Roy and Hasan, 2018). A successful evacuation highly depends on real-time evacuation route guidance and traffic management (Pel et al., 2012). These activities rely on how accurately we can estimate and predict traffic states in real-time. Thus, reliable predictions of travel time will allow people to make an informed decision on whether to evacuate or not. Moreover, this will allow emergency management authorities to decide whether to order an evacuation or not. For instance, during hurricane Harvey, evacuation orders were not widely issued due to the fear of massive traffic congestion, potentially causing loss of lives during evacuation (Dave, 2017). In addition, reliable predictions of future traffic will enable transportation agencies to react proactively during a hurricane evacuation.

Several data-driven methods have already been used in classification and prediction related problems. One of the major benefits of data-driven methods is easy to deploy in a real-time context. Few commonly used data-driven approaches are Support Vector machine, K-*nearest neighbor*, Analytic Neural Network, ARIMA, etc. These models performed reasonably well for predicting traffic states under regular traffic demand (Billings and Jiann-Shiou, 2006; Lee, 2009; Myung et al., 2011). But under irregular traffic demand, we have to deal sharp non-linearities in traffic flow patterns over time. Therefore, traditional prediction models may not work well in such conditions. To overcome this challenge, deep learning techniques can be a viable solution. It is a machine learning technique that uses non-linear functions (tanh, logistic etc.) to capture the sharp discontinuities in traffic flow (Polson and Sokolov, 2017).

In this study, we present a method to predict the time mean speed of freeways. We adopt a deep learning technique known as Long-Short Term Neural Network and assess its performance against the existing data-driven approaches. We have compared the performance of the LSTM-NN model with Auto-Regressive Integrated Moving Average (ARIMA), K-Nearest Neighbor Regressor and Analytic Neural Network. For this study, we have collected traffic data from I-75 which was a major evacuation route for Hurricane Irma.

3.2 Existing Works

With the advancement of sensor technologies, short-term traffic forecasting has become a critical component for Intelligent Transportation Systems. It predicts traffic states for few seconds to few hours ahead of current time (Vlahogianni et al., 2014b). Previous studies focused on methods to model traffic characteristics such as volume, density, and speed or travel time. These approaches can be broadly classified into three categories: model-driven, data-driven, and streaming data-driven (hybrid) (Seo et al., 2017). Model-driven approaches can be further classified into two levels macroscopic and microscopic. Macroscopic models rely on the fundamental relationship among different parameters (flow, density, speed) of traffic flow rather than individual vehicles. While Microscopic models focus on a single vehicle or intersection (or a small number of intersections). For traffic state estimation, microscopic models rely on data available from signal timing, vehicle counts or high penetration rate travel time measurements (Ban et al., 2010).

Sometimes it is tedious to gather detailed parameters required for a model-driven approach; hence recent studies are exploring alternative data-driven approaches.

A data-driven approach relies on historical traffic patterns to estimate future traffic states. It does not consider the influence of traffic flow mechanism on traffic dynamics (Oh et al., 2017). Several data-driven parametric and non-parametric approaches have been used for short-term traffic state prediction. Among the parametric models, ARIMA (Billings and Jiann-Shiou, 2006) has performed better than other parametric time series prediction models. Researchers have also explored non-parametric models for improving prediction accuracy including Kalman Filter (Chu et al., 2005), Support Vector Machine (Ahn, 2016; C. Wu et al., 2004), K-Nearest Neighbor (Cai et al., 2016; Habtemichael and Cetin, 2016; Meng et al., 2015; Myung et al., 2011; Qiao et al., 2013; Yu et al., 2016), and Artificial Neural Network (Innamaa, 2005; Lee, 2009; Park et al., 1999; Yu et al., 2008) .

Hybrid models combine data-driven and model-driven approaches. For instance, Hofleitner et al. (Hofleitner et al., 2012) implemented a hybrid model integrating hydrodynamic theory of traffic flow with a Bayesian network approach. They derived an analytical probability distribution of travel times between arbitrary locations using kinematic wave theory.

Recent developments in computational techniques allow us to overcome different challenges in developing an effective prediction system. Vlahogianni et al. (Vlahogianni et al., 2014b) discussed several challenges, such as a system's characteristics which integrate prediction models, choosing appropriate variables while forecasting, modeling issues related to developing effective prediction algorithms, role of artificial intelligence models and how they will be integrated with prediction schemes.

One of the major challenges for predicting traffic state is the presence of sharp non-linearities due to transitions among free flow, breakdown, recovery, and congestion (Polson and Sokolov, 2016). Recently, deep learning techniques have been used to capture such nonlinearities. Duan et al. (Yanjie Duan et al., 2016) applied a deep learning model, LSTM neural network which is an advanced version of Recurrent Neural Network for travel time prediction. They have constructed 66 series prediction LSTM neural networks for the 66 links in the dataset. Ma et al. (Ma et al., 2015) also used LSTM neural networks to predict speed using only two microwave detectors data (speed and volume). In both studies, they did not consider the influence of temporal (time of the day, the day of the week) variation on prediction accuracy. Another important consideration is that the traffic state (speed, volume, etc.) of a particular roadway link depends on the upstream and downstream link traffic state but they have not considered this influence as well. Moreover, they have not tested the performance of these models under irregular traffic conditions (such as hurricane evacuation period or any other events). Although Cui et al. (Cui and Wang, 2017) have proposed a deep stacked bidirectional and unidirectional LSTM-NN, which considers both backward and forward dependencies of time series data, to capture spatial and temporal dependencies from the historical data; but they have not evaluated their model performance during irregular traffic demand.

In our study, we consider temporal variations and spatial relationships among the upstream, target, and downstream links. We evaluate our model's performance under an extreme traffic condition. Our model performs better than other state-of-the-art approaches which indicate the potential of LSTM-NN model in time series prediction. A critical issue for adopting a data-driven approach is the required computation time. We adopt the tensor flow library (Abadi et al., 2016); it allows us to break the training process into several chunks and run them in parallel across multiple CPUs or GPUS within a reasonable amount of time. This makes it possible to train a network with millions of parameters on a training set composed of billions of instances (Geron, 2017).

3.3 Framework for Speed Prediction

In this study, we assume that for a particular link, the average spot speed at a given time step (t) will depend on the average spot speed of the upstream and downstream links adjacent to this link. So, to formulate the modeling framework we have added the upstream ($X(t)=S_{m-1}(t)$), Downstream ($X(t)=S_{m+1}$ (t) and Target link traffic state (X (t)=Sm) as input vectors to predict the target link speed after 5min, 10 min, and 15 min time interval (Figure 3.1).

Moreover, to capture this temporal influence we added the time of the day and day of the week as independent variables. In a regular traffic scenario, we can observe that the daily variation of speed and volume follows a recurrent pattern for example, at the morning and evening peak hour traffic volume is higher, which means the overall speed at this time period is lower. Similarly, traffic flow patterns are different on both weekdays and weekends. In case of weekdays, traffic volume is quite higher than the weekends. So, overall speed of the vehicles is lower. In our case, we are considering an irregular traffic demand (hurricane evacuation) scenario where traffic pattern is non-recurrent. Hence, we cannot apply the same assumption for both regular and irregular scenario. But we have to maintain uniform modeling framework for both regular and irregular traffic demand. Hence, we need an approach which will be able to capture the regular behavior as well as irregular behavior by learning long-term and short-term dependencies among different traffic states over time. This framework is developed to check whether LSTM NN model can serve this purpose.

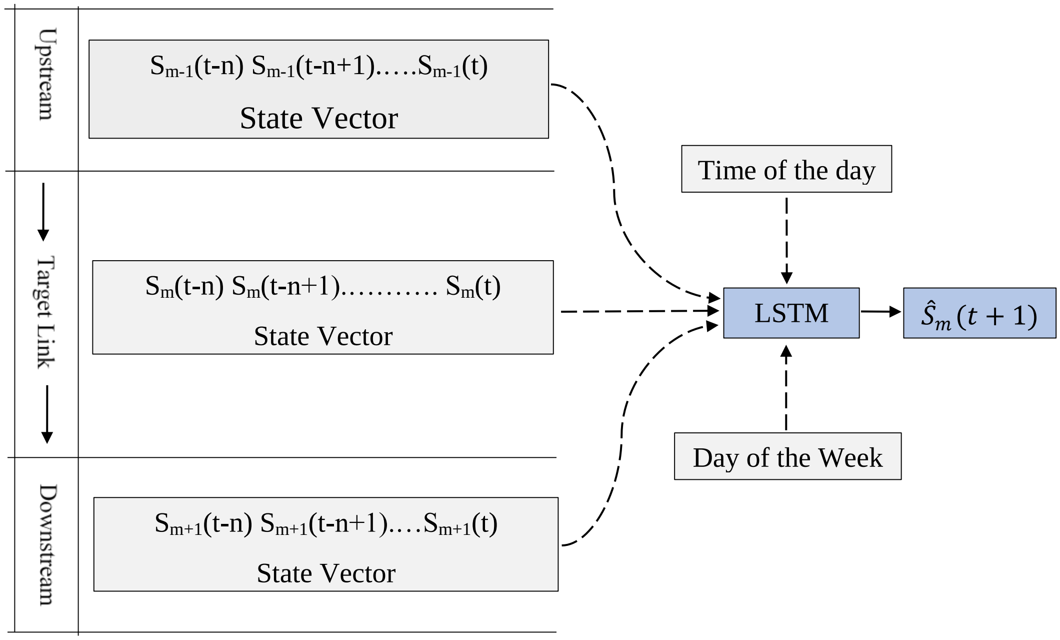
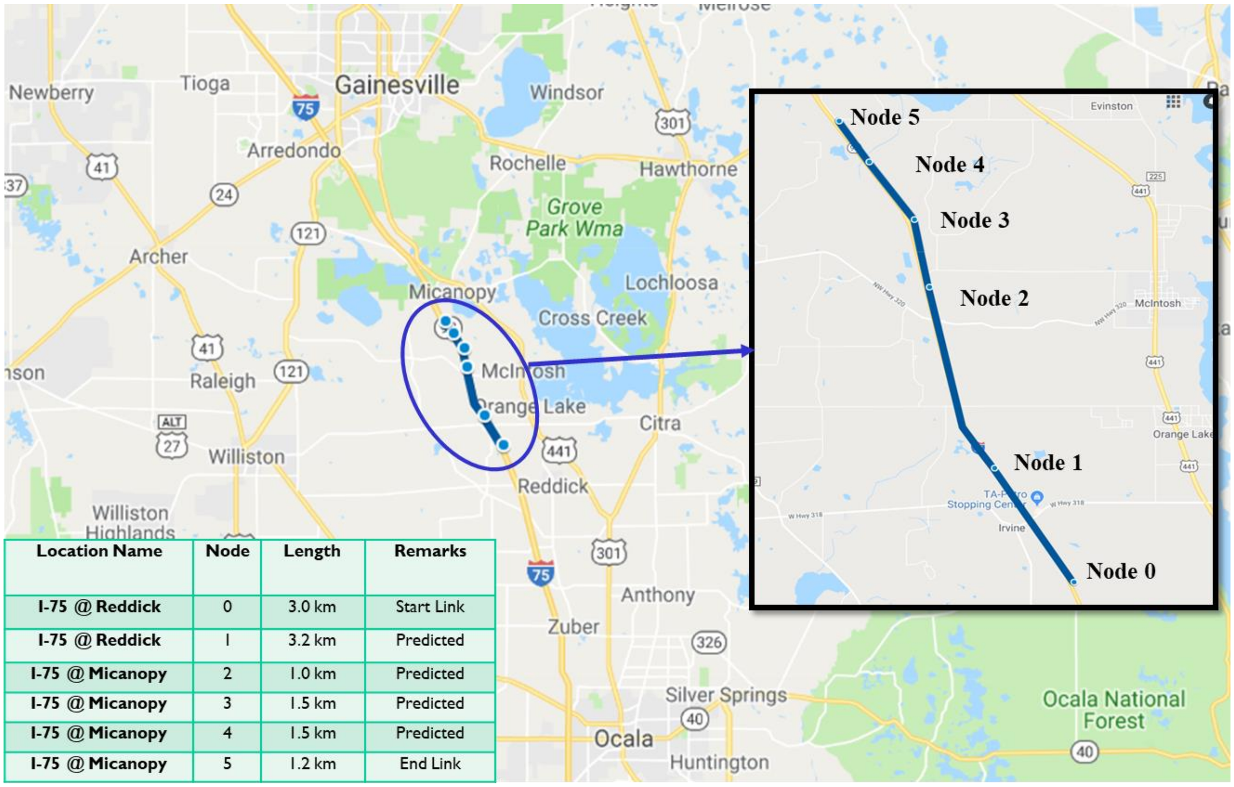


Figure 3.1: The Layout of the Variables for Prediction

3.4 Case Study

3.4.1 Study Location

One of the primary objectives of this study is to evaluate the performance of the LSTM-NN model in case of irregular traffic demand, such as during a hurricane evacuation. To do so, we collected the data for 11.4 km long segment of the I-75 from September 3, 2017, to September 17, 2017. This time span covers the evacuation period of hurricane Irma. To select the study location, we observed previous evacuations to understand major evacuation routes . Observing the evacuation pattern from historical data, we found that a large portion of residents living in Florida evacuates to Georgia or adjacent States (Roy and Hasan, 2019). Hence, we have chosen a location between Ocala to Gainesville, a road segment which had to serve a major portion of the evacuation traffic during Irma. We have collected data from six MVDS detectors (Figure 3.2); each detector provides real-time speed and volume. For this study, we have used an average of the time mean speed over a five-minute interval.



Location Name	Node	Length	Remarks
I-75 @ Reddick	0	3.0 km	Start Link
I-75 @ Reddick	1	3.2 km	Predicted
I-75 @ Micanopy	2	1.0 km	Predicted
I-75 @ Micanopy	3	1.5 km	Predicted
I-75 @ Micanopy	4	1.5 km	Predicted
I-75 @ Micanopy	5	1.2 km	End Link

Figure 3.2: Study Location I-75 (Google Map,2018)

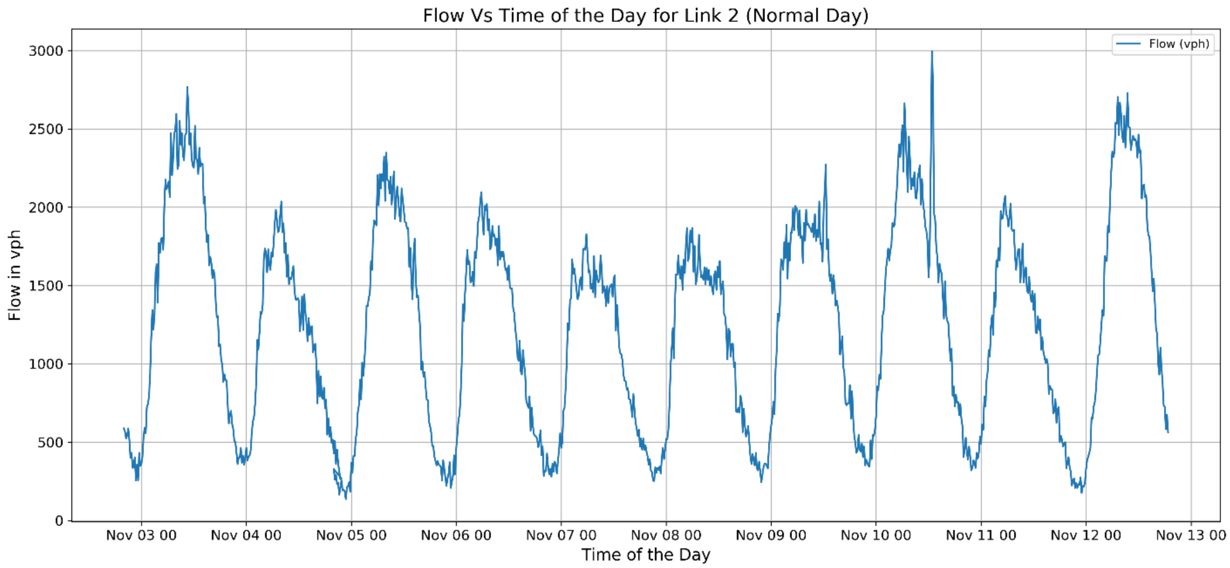
To compare the prediction accuracy of LSTM NN model for regular and irregular demand scenario we also collected the traffic data for the same location for non-evacuation period from November 03, 2017 to November 17, 2017.

3.4.2 Data Exploration

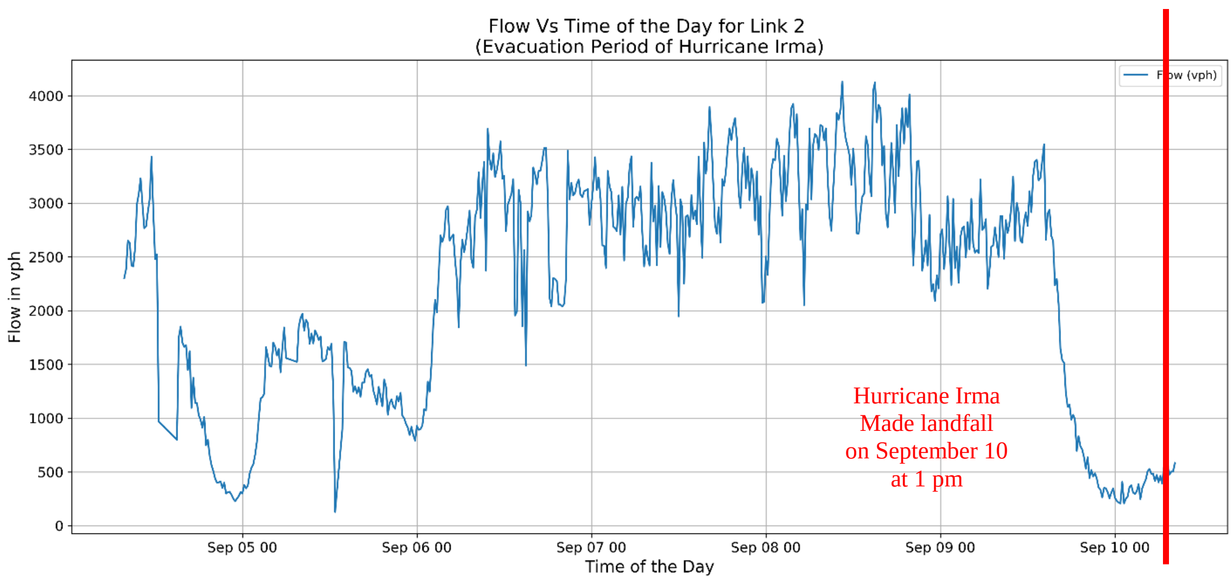
We analyzed both regular and evacuation traffic data, from our analysis we observe a regular traffic pattern during normal traffic condition. We analyzed the northbound traffic of I-75, hence we can observe morning peak in between 8 to 10 am (Figure 3.3 (a)). But during the evacuation period, there is no regular pattern (distinctive evening peak) over time (Figure 3.3(b)). Moreover, there is a heavy volume of traffic especially in the period from September 06, 2017 to September 09, 2017 (close to the landfall day). Hence overall flow was quite higher than the regular.

Hurricane Irma made its landfall at the Florida Keys on September 10, 2017, at category 4 intensity; then it passed over several regions of Florida in between September 10, 2017, to September 12, 2017, and caused a power outage at several locations. It took about a week to restore the overall system. That is why we were unable to collect data between September 11, 2017, to September 16, 2017.

Figure 3.4 (a) shows an irregular variation of speed over time. This is because of the high volume of traffic, particularly on September 7^{th} to onwards due to the evacuation of a large number of people from Florida to other locations. So, travel time at this period was quite higher than the regular time.

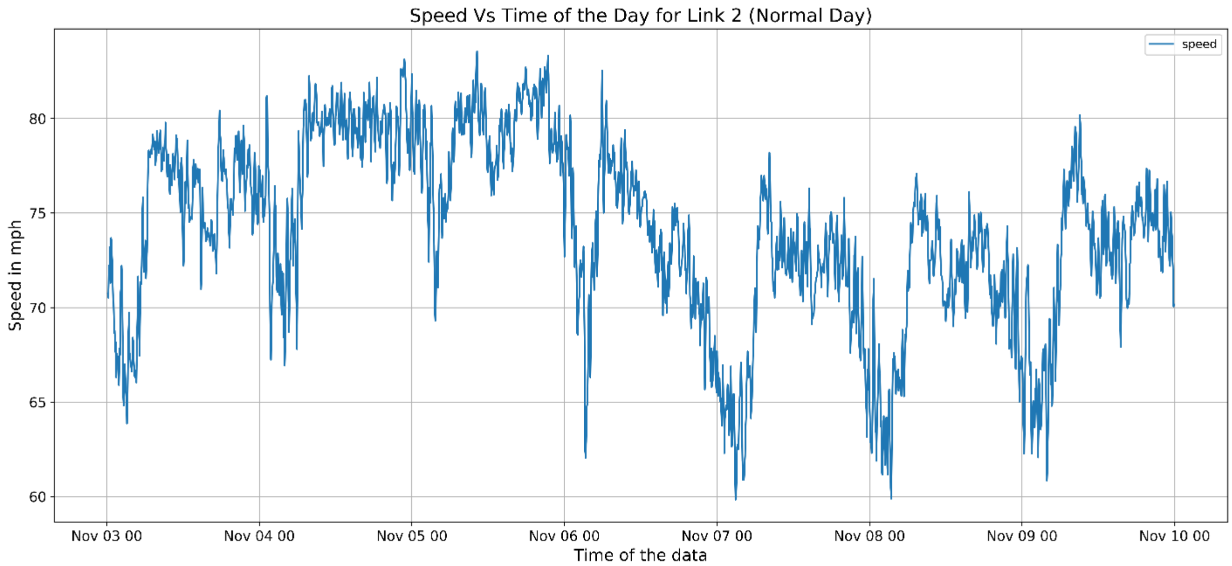


(a)

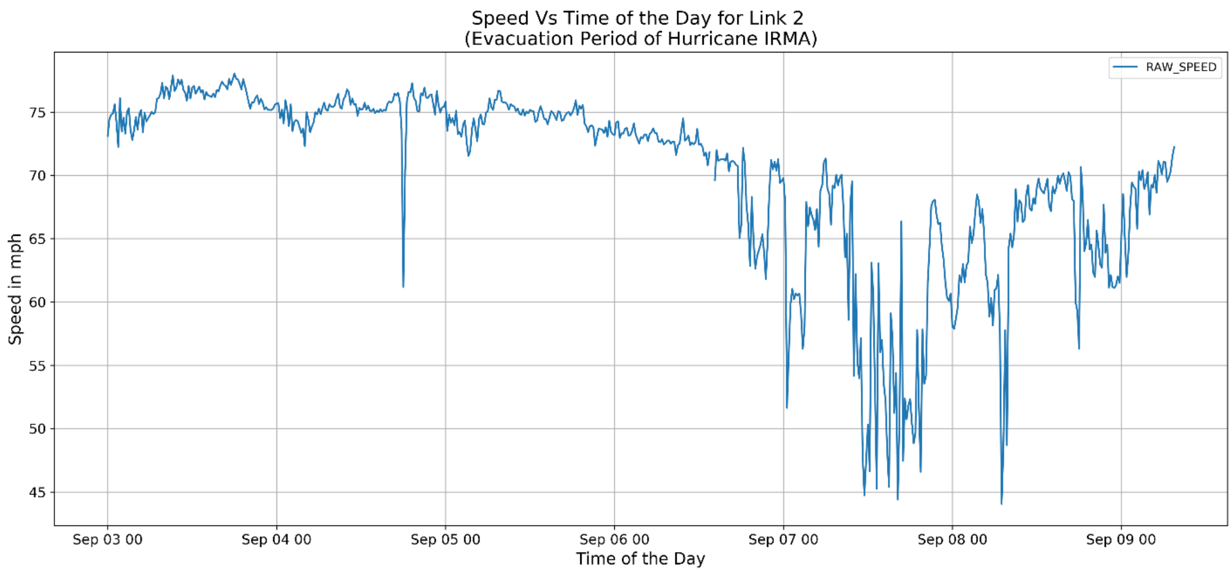


(b)

Figure 3.3: Variation of Flow with Time of Day (a) Normal day (Nov. 03, 2017 -Nov. 13, 2017) (b) Evacuation Period of Hurricane Irma (Sept 03, 2017 - Sept 13, 2017)



(a)



(b)

Figure 3.4: Variation of Speed With Time of the day (a) Normal day (Nov. 03, 2017 -Nov. 13, 2017) (b) Evacuation Period of Hurricane Irma (Sept 03, 2017 - Sept 13, 2017)

3.4.3 Model Training

The flexibility in deep neural networks has created a major challenge to select the combination of hyperparameters that will work best for a certain task. To solve this issue several methods have been developed such as grid search, random search, Bayesian optimization or sequential model-based optimization (SMBO) (Bergstra et al., 2013, 2011; Hutter et al., 2011). In this study, we applied SMBO with tree-structured parzen estimator (TPE) algorithm to obtain the best combination of hyperparameters. SMBO methods sequentially construct models to approximate the performance of hyperparameters based on historical measurements, and then subsequently choose new hyperparameters to test based on this model. SMBO methods work best for scalar-valued functions which are costly to evaluate compared to conjugate gradient descent methods and model-based optimization algorithms.

We implemented the SMBO optimization method using hyperopt library(Bergstra et al., 2013). The hyperopt library gives the ability to define a prior distribution for each parameter. Table 3.1 shows the information regarding the parameters that we are going to tune. To evaluate the best performing model, we use mean squared error as a loss function.

Table 3.1: Prior Distribution of Each Parameter for Speed Prediction

Parameter Name	Distribution	Values
Number of Hidden Layers	Categorical	$x \in \{1,2\}$
Activation Function in each layer	Categorical	$x \in \{relu, sigmoid, tanh\}$
Number of Units in First Layer	Categorical	$x \in \{64,128,256\}$
Number of Units in Second Layer	Categorical	$x \in \{64,128,256\}$
Dropout in each layer	Uniform	$x \in [0,1]$
Optimizer	Categorical	$x \in \{adam, sgd, adagrad, rmsprop\}$
Batch Size	Categorical	$x \in \{12,24,48,72,96,144\}$

To predict future traffic speed, we have divided the dataset into a training and a test set. Data from the first 5 days (Nov. 3, 2017 – Nov. 7, 2017) is used for training the model and the rest 2 days (Nov. 8, 2017- Nov. 9, 2017) data is used for validation. We ran the SMBO algorithm on different datasets corresponding to different roadway segments (four target links) and different prediction horizon (5 min, 10 min, 15 min), finally, we obtain the optimal combination of hyperparameters which works best for each dataset.

While training the LSTM NN model we do not pass entire dataset rather we divide the dataset into small batches. Hence, at each iteration, the model learns the entire dataset in small batches and then move into the next iteration and do the same. As shown in Table 3.1 we choose categorical distribution of batch size over $\{12,24,48,72,96,144\}$. From the SMBO algorithm, we found that the model works best for a batch size of 72. Table 3.2 shows the optimal parameters for the final LSTM-NN model.

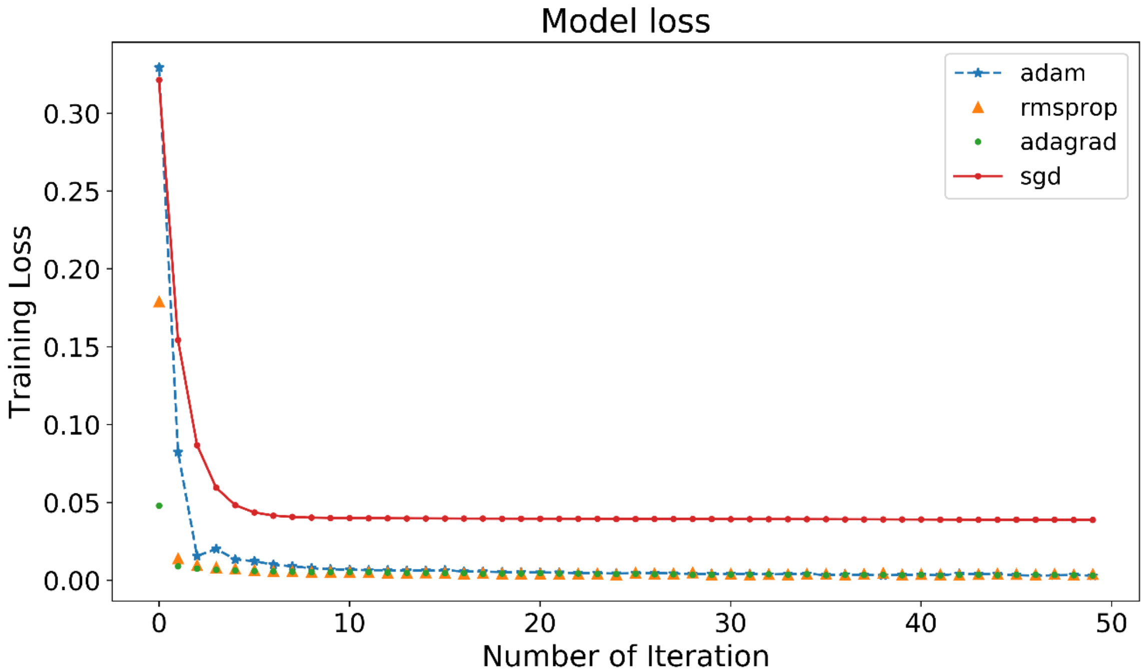


Figure 3.5: Variation of Training Loss per Iteration for Different Optimizer (Batch Size =72)

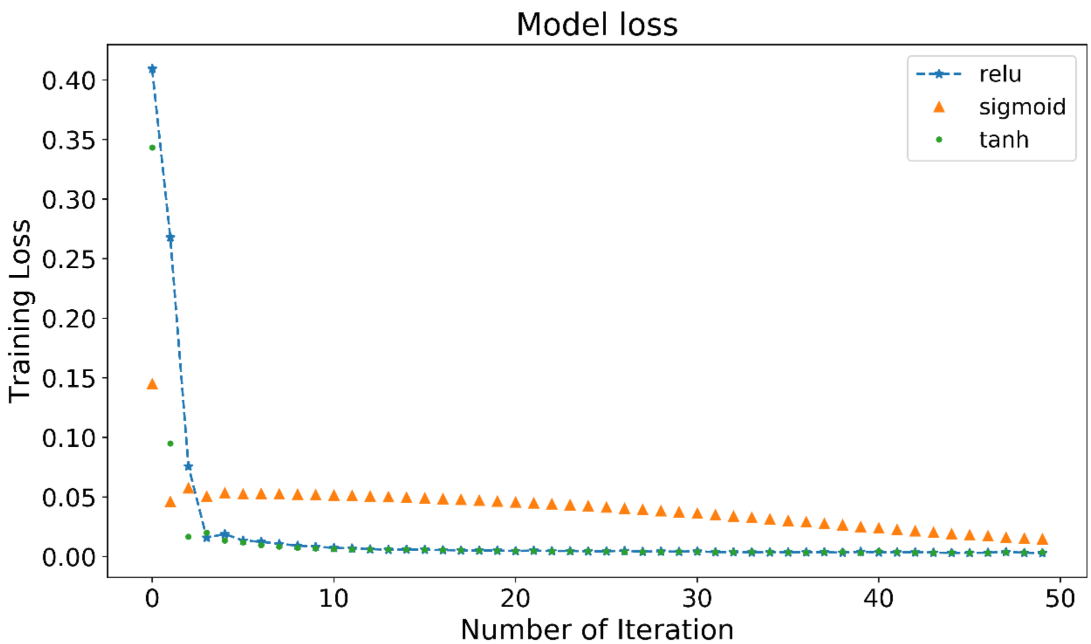


Figure 3.6: Variation of Training Loss per Iteration for Different Activation Function (Batch Size =72, Optimizer = adam)

From the optimization result, we found that adaptive moment estimation (adam) optimizer works better than root mean square propagation (rmsprop), adaptive gradient (adagrad) and stochastic gradient descent (sgd) optimizer (Figure 3.5). At the same time, it converges faster than the others and takes less time to train the model. Figure 3.6 shows the training loss for different activation function. Both relu and tanh activation function work better, but if we choose sigmoid function the model starts overfitting at certain points before converging to the validation loss. Hence, we need to add large dropout at each layer to control the training process and it takes a long time to converge.

The dropouts are added to control overfitting of the training set. But for our case the dropout value is so small if we ignore these values (dropout =0), it does not affect the model performance. We also applied the early stopping criteria to avoid overfitting. The model stops training when training loss is less than the validation loss. Figure 3.7 shows the training and validation loss for the best model. We can see that the model converges after 15 iterations (epoch).

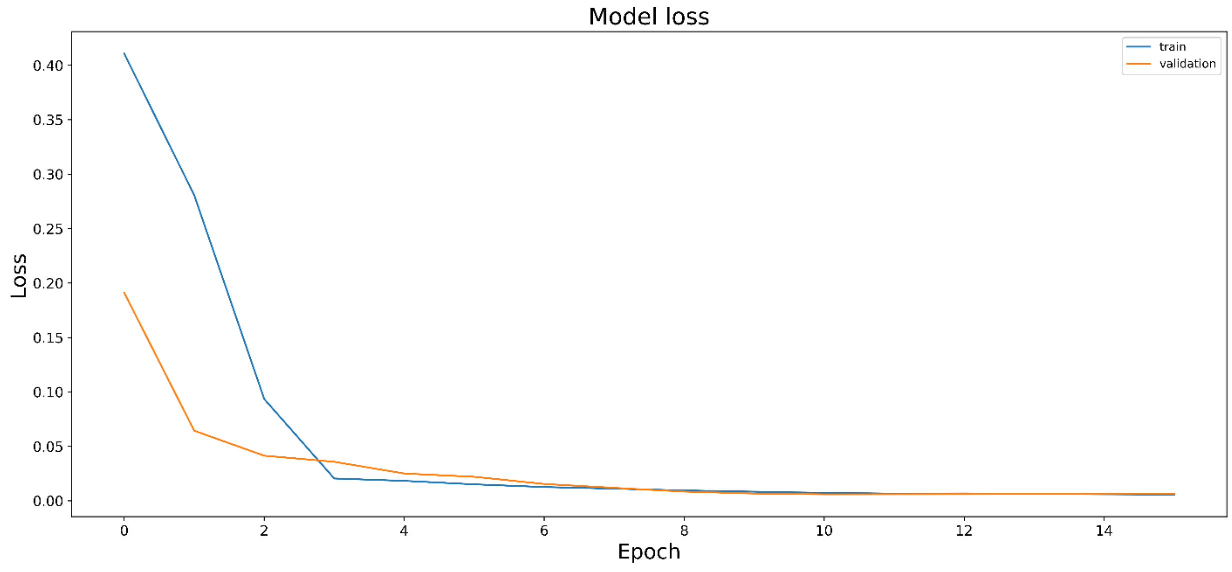


Figure 3.7: Training and Validation Loss for the Optimized model

Table 3.2: Hyperparameters for best Performing Model for Speed Prediction (Normal Day)

Number of Hidden Layers	Number of Hidden Units	Dropout	Activation Function	Optimizer
First	128	0.002	relu	Adam
Second	64	0.001	relu	

Table 3.3: Hyperparameters for best Performing Model for Speed Prediction (Evacuation Period)

Number of Hidden Layers	Number of Hidden Units	Dropout	Activation Function	Optimizer
First	128	0.1	tanh	Adam
Second	64	0.05	tanh	

We did the same experiment with the hurricane evacuation traffic data. we train the LSTM NN model to learn the patter of the data during a hurricane evacuation. We use the data from Sept. 3, 2017 – Sept. 7, 2017, for training the model and the rest 2 days (Sept. 8, 2017- Sept. 9, 2017) data is used for validation. Table 3.3 shows the selected hyperparameter for the trained model for the evacuation traffic data.

3.4.4 Experimental Results

We have calculated Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE) as performance measures to check the accuracy of the implemented model. Performance metrics are defined as,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} \tag{7}$$

$$\text{MAE} = \frac{\sum_{t=1}^{n}|y_t - \hat{y}_t|}{n} \tag{8}$$

$$\text{MAPE} = \sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{y_t} \times 100\% \tag{9}$$

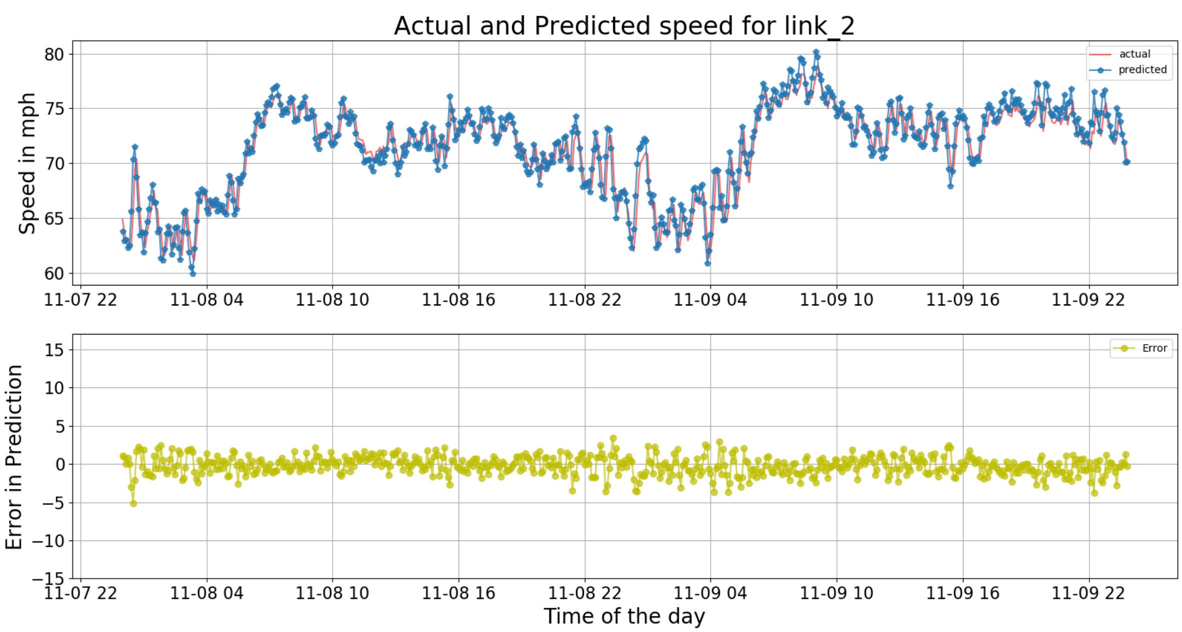


Figure 3.8: Actual and Predicted Speed and their difference (the x-axis is divided into 6-hour intervals; 11-08 04 means Nov. 08, 4 AM)

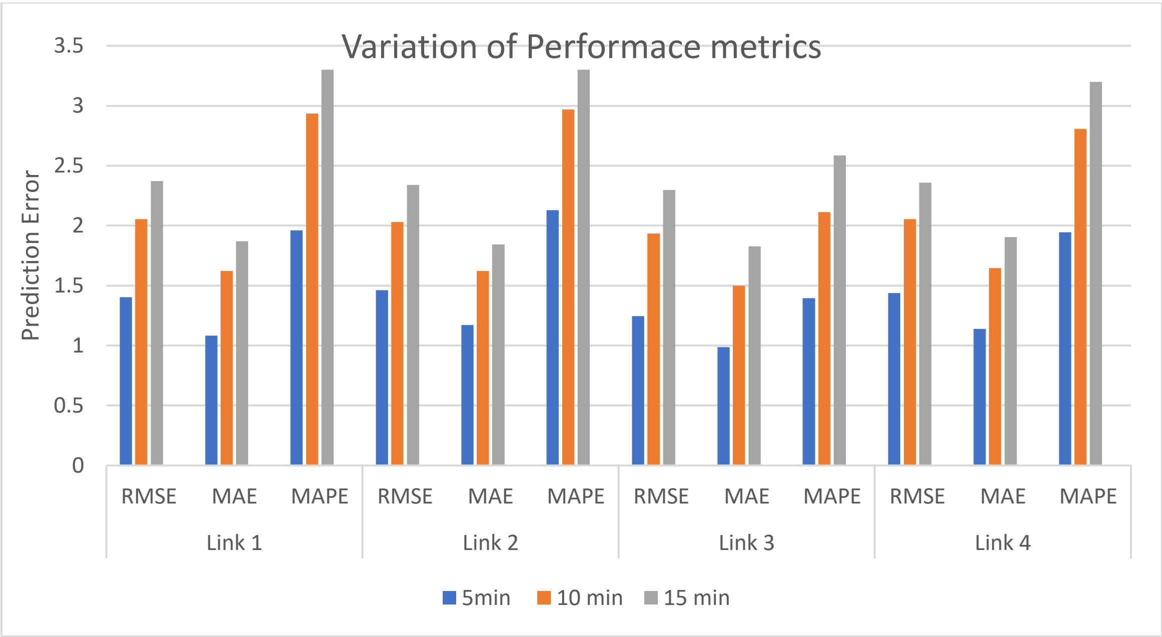


Figure 3.9: Variation of Performance Metrics with Prediction Horizon

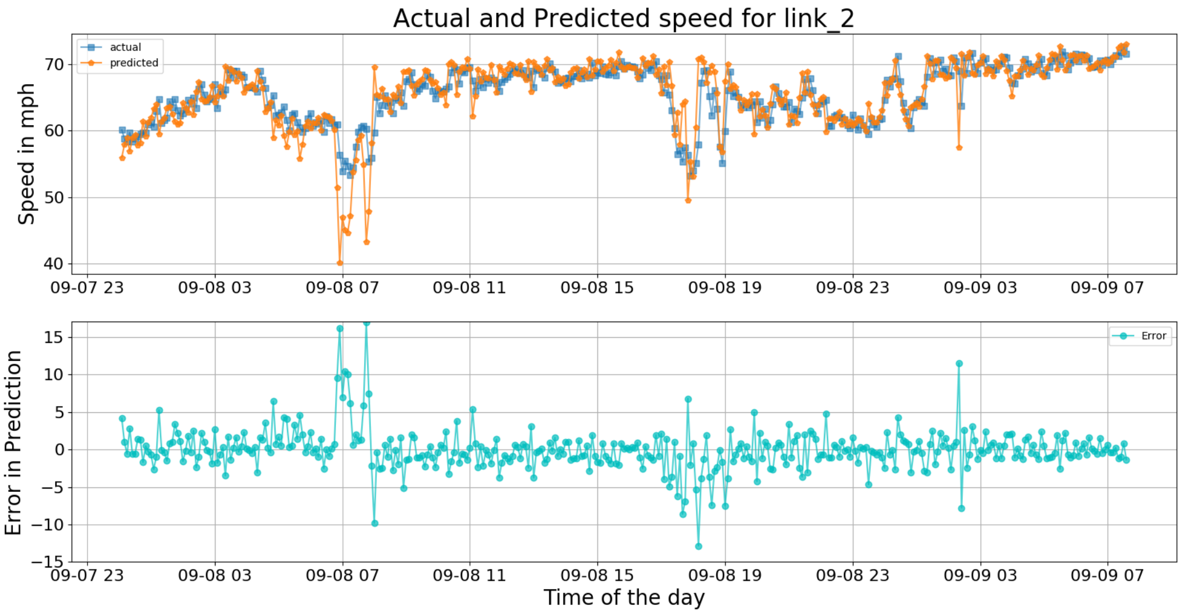


Figure 3.10: Actual and Predicted Speed and their difference (the x-axis is divided into 6-hour intervals; 09-08 02 means Sept. 08, 2 AM)

Figure 3.9 shows the variation of actual and predicted speed for the second link under normal traffic condition. The difference between actual and predicted speed is quite low. As shown in figure 3.9 the RMSE and MAE values varies in between 1 to 2 for different links and prediction horizon. The maximum MAPE value is 3.2, which means the least accuracy of the model in around 97%.

Figure 3.10 shows the difference between actual and predicted speed for evacuation traffic data. Surprisingly, the difference between actual and predicted speed is quite low even though the traffic flow variation is irregular during this time period. Which indicates that LSTM-NN has captured the nonlinearities well. Moreover, the RMSE value for the target links varies between 2 and 4 (Figure 3.11) while the MAE values vary between 2 and 3. Thus, LSTM-NN model performs better even in the case of irregular traffic demand, indicating its effectiveness in capturing nonlinearities.

We also compare the LSTM-NN model with the most popular time series model ARIMA and two other commonly used data-driven models KNN and ANN over multiple performance metrics. Figure 3.11 shows that the accuracy level (based on MAPE) for LSTM varies between 96 and 97% except in case of Link 4 where RMSE value is found 5.11 while predicting 15 min ahead of current time (Figure 3.11). For ANN, KNN, and ARIMA accuracy levels vary as 94-96%, 93-94%, and 88-93%, respectively. Moreover, the MAE and RMSE values for LSTM-NN are quite low compared to other models. From the result, we can conclude that the neural network models (LSTM-NN, ANN) can learn the sequential time series data better than others. However, considering each matrices LSTM-NN performs better.

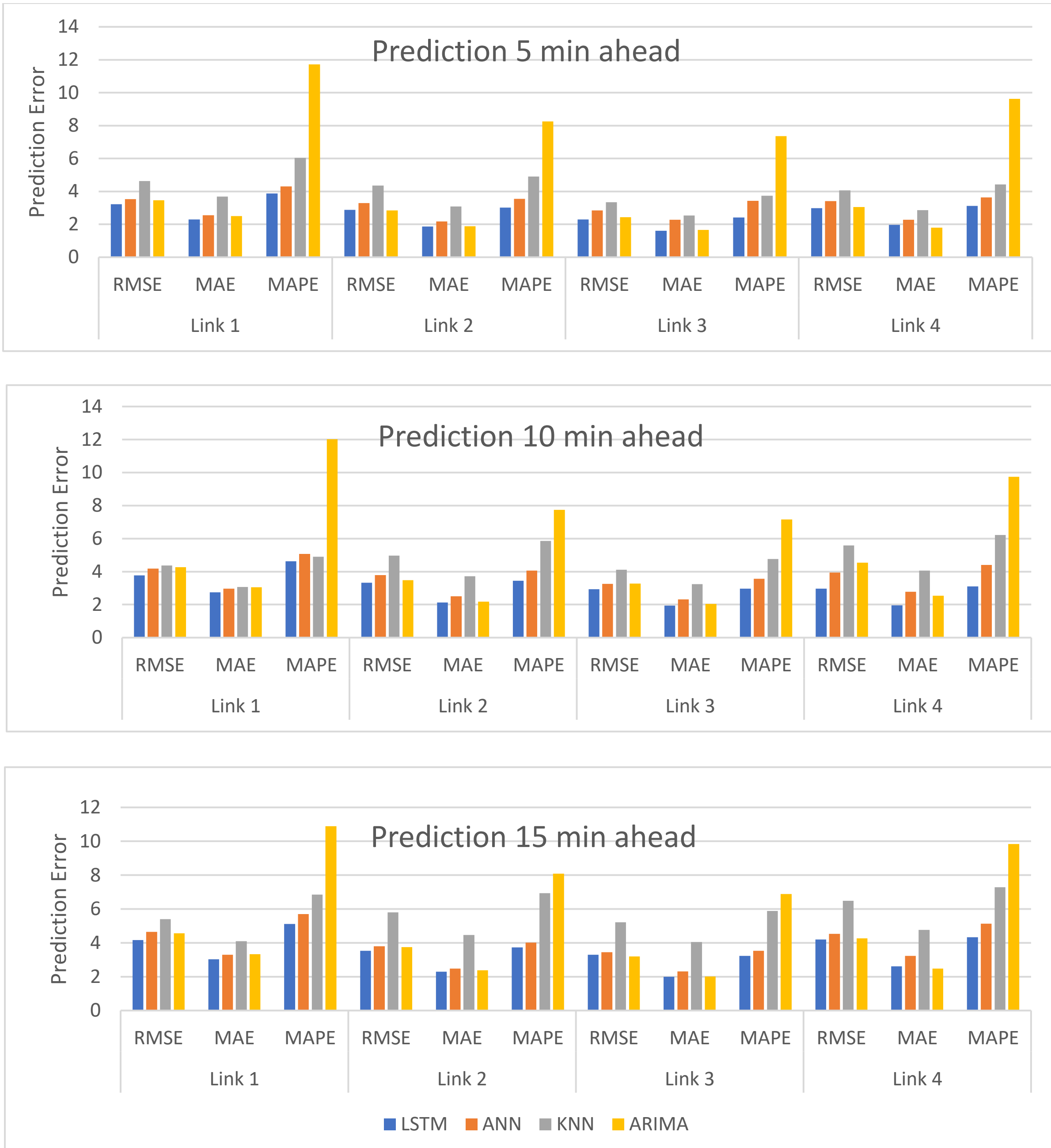


Figure 3.11: Comparison of LSTM Model based on performance Indexes

3.5 Discussion

This study focuses on predicting time mean speed of freeways using LSTM NN model considering temporal and spatial dependency of the traffic data. We developed a modeling framework considering the fact that, the future speed of a particular link depends on its upstream and downstream link speed as well. We did the experiment for both regular and irregular traffic demand condition. In both cases, the model performed reasonably well. This indicates the effectiveness of LSTM-NN model in capturing nonlinear relationships among traffic variables. We have compared the performance of LSTM NN model with other traditional models and found that it outperforms both parametric and non-parametric models. However, each of the model (KNN, ANN, LSTM) performed reasonably well, which means our modeling framework can capture the spatial and temporal relationships among traffic variables with better accuracy.

CHAPTER 4: TRAFFIC SIGNAL QUEUE LENGTH PREDICTION

4.1 Introduction

Inefficiencies in traffic signal timing due to poor green time allocation, inability to respond quickly to real-time conditions, and lack of coordination between adjacent intersections are a few major causes of congestion problem (Smith et al., 2013). Researchers from multiple fields are testing innovative traffic control systems that can effectively manage traffic in a signal based on real-time traffic flows. Adaptive Traffic Control System (ATCS) is a state-of-art-traffic control system and a major component of the intelligent transportation system (ITS) which can efficiently manage and distribute traffic in real-time.

ATCS technologies gather information regarding current traffic demand and use it to optimize different parameters of a traffic controller (e.g., cycle length, split, offset, and phase sequence depending on the system) (FDOT, 2016). One of the main performance measures of the ATCS is queue length, which also plays a crucial role in signal optimization. Current adaptive signals mostly rely on infrastructure-based sensors or video-based loop detectors to estimate the queue length. Using these detectors have several limitations: they only provide instantaneous position of a vehicle rather than direct measurement of traffic (speed, location) states; the installation and maintenance cost of the detection system is considerably high (Feng et al., 2015); and they estimate queues that are shorter than the distance between vehicle detector and intersection stop line (Liu et al., 2009). Moreover, if one or more loop detectors start malfunctioning, the performance of the adaptive signal control system worsens significantly.

In this study, we consider a corridor of intersections where consecutive intersections will share information with each other and gather information of upcoming vehicles. We develop a data-driven approach to predict the lane-based queue length for an intersection. We anticipate that with emerging connected vehicles technologies and road environments, information (traffic state, queue length etc.) from one intersection will be easily available to another intersection. For our experiments, we used InSync Adaptive signal data which provides queue lengths and wait times (time required for the first vehicle to clear the intersection) for different vehicular movements. We trained a Long-Short Term Memory Neural Network (LSTM-NN) model to predict the queue length for the next cycle based on queue length and wait time of three consecutive intersections at the current cycle. We run the experiments to predict queue lengths for north through traffic. The same methodology can be applied to predict queen lengths for other movements as well.

4.2 Existing Works

Vehicular queue length estimation is crucial in optimal signal planning (Chang and Lin, 2000; Mirchandani and Zou, 2007; Newell, 1965) as well as measuring signal performance for a signalized intersection (Balke et al., 2005). Especially for ATCS technologies, the signal control logic is based on real-time estimated queue lengths. So far, a vast amount of works has been done in this field and researchers have already developed several methods to estimate queue lengths for traffic signals using loop detector data and signal timing information. These studies can be classified into two categories. The first one is based on the analysis of cumulative input-output to a signal link which was proposed by Webster in 1958 (Webster, 1957), later improved by several researchers (May, 1975; Newell, 1965; Robertson, 1969; Sharma et al., 2007; Vigos et al., 2008). In this method, the queue length is derived from cumulative arrivals and departures of an intersection. However, this model is effective in describing the queue length formation process or effective queue size, but not sufficient to obtain the spatial distribution of queue length for a given time (Stephanopoulos et al., 1979). Moreover, the application of this approach is limited, since cumulative input-output methods can be applied only when the queue length does not exceed the vehicle detector location (Liu et al., 2009). The second category is based on shockwave analysis: how queue forms and dissipates at an intersection. Lighthill, Whitham (Lighthill and Whitham, 1955) and Richards (Richards, 1956) first demonstrated this theory for uninterrupted flow. Stephanopolos and Michalopoulos (Stephanopoulos et al., 1979) expanded it for signalized intersections.

With the advancement in vehicle detection and sensing technologies, it has become easier to collect multi-resolution traffic data. Consequently, real-time queue length estimation such as cycle by cycle queue length has gained more attention. Several studies have been conducted using event-based signal and high-resolution loop detector data (An et al., 2017; Balke et al., 2005; Liu et al., 2009; Smaglik et al., 2007) for real-time queue estimation. Moreover, mobile traffic sensors, such as GPS equipped probe vehicles, cellular phones, connected vehicles, and other tracking devices, provide a supplement or alternative to fixed-location sensors for real-time queue estimation. GPS equipped probe vehicle data have shown great potential for real-time queue length estimation (Comert, 2013; Hao and Ban, 2015; Jeff Ban et al., 2011). Connected vehicle technologies have created new opportunities for queue length estimation, Tiaprasert et al. (Tiaprasert et al., 2015) presented a mathematical model for real-time queue estimation using connected vehicle technology for adaptive signal control.

Even though high-tech sensing devices and connected vehicle technologies creating great opportunities to get multiresolution traffic data, but data-driven queue length estimation techniques are less common. Chang and Su (Chang and Su, 1995) were the first to explore the data-driven neural network model for predicting queue length at a short time step (3s). They used extensive data from simulation experiments and created multiple scenarios to experiment with the model. The prediction accuracy of the model was more than 90% at 3-time steps. However, in this study, we have used a different approach by applying a deep LSTM-NN model to capture the long-term dependencies of the traffic flow pattern. Moreover, we have considered a connected corridor with multiple intersections rather than a single intersection.

4.3 LSTM-NN Architecture for Queue Length Prediction

In this study, we assume that for a given intersection, the queue length for a specific movement will depend on that intersection and upstream intersections. For example, north through (NT) for the next cycle (t+1) will depend on the queue length and vehicle wait time of that intersection and the adjacent upstream intersections at current cycle (t). As input vectors, we have added the upstream intersections and target intersection queue length and wait time ($X(t)=[q_{i-2}(t), q_{i-1}(t), q_i(t)$ $w_{i-2}(t), w_{i-1}(t), w_i(t)]$) (Figure 4.1).

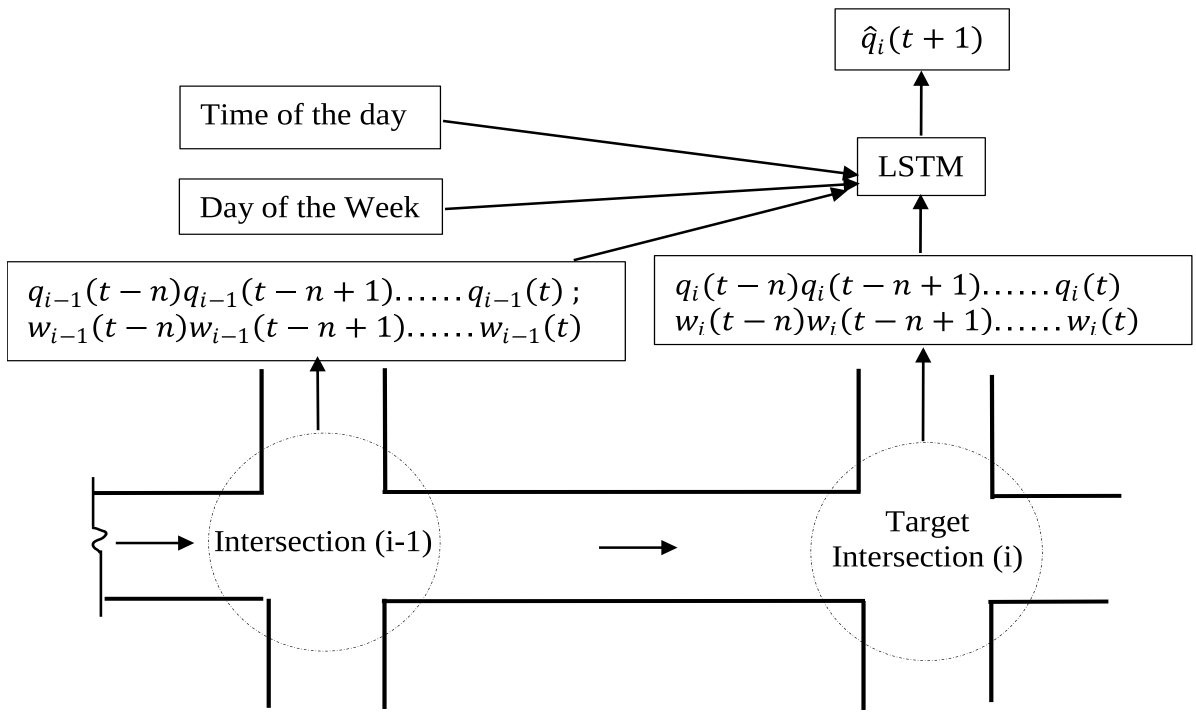


Figure 4.1: The layout of the Variables for Prediction

Moreover, to capture this temporal influence we added the time of the and day of the week as independent variables. In a regular traffic scenario, we can observe that the daily variation of traffic flow follows a recurrent pattern. For example, in the morning and evening peak hour traffic volume is higher, which means the overall speed at this time period is lower. Similarly, traffic flow patterns are different on both weekdays and weekends. In case of weekdays, traffic volume is quite higher than the weekends.

4.4 Case Study

4.4.1 Data Description

For this study, we collected adaptive traffic signal data from InSync between December 18, 2017, and February 14, 2018. We collected the data for the corridor of Alafaya Trail (SR-434) located in East Orlando, FL, from its Waterford lake intersection to McCulloch road intersection including 11 intersections in total (Figure 4.2). InSync database provides mainly two types of data: (i) Turning Movement Counts (TMC) - vehicle counts per phase and lane for every 15 minutes; (ii) History data which provide the details of each movement with the time, duration, queue and wait time (refers to the wait time in seconds of the first car that was detected on the phase at the time logged) for each phase. In general, the history data contains information regarding eight distinct movements North Left (NL), North Through (NT), South Left (SL), South Through (ST), East Left (EL), East Through (ET), West Left (WL) and West Through WT). Movements of pedestrians, bicycles or any non-motorized vehicle are considered as a separate phase.

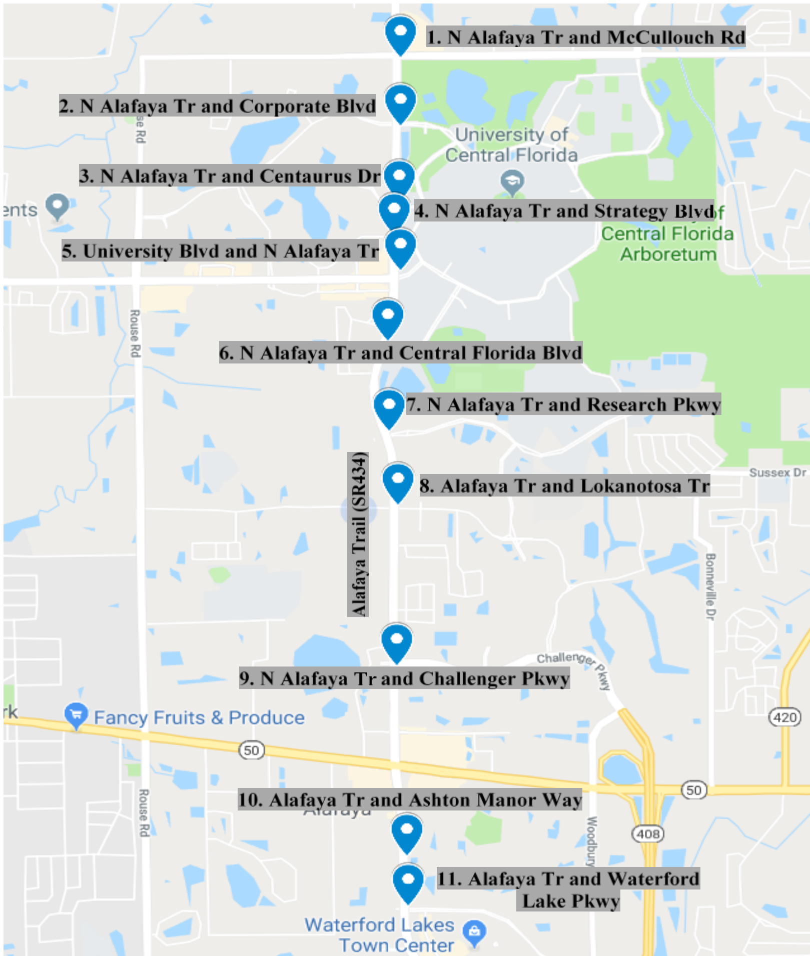


Figure 4.2: Study Location (Google Map, 2018)

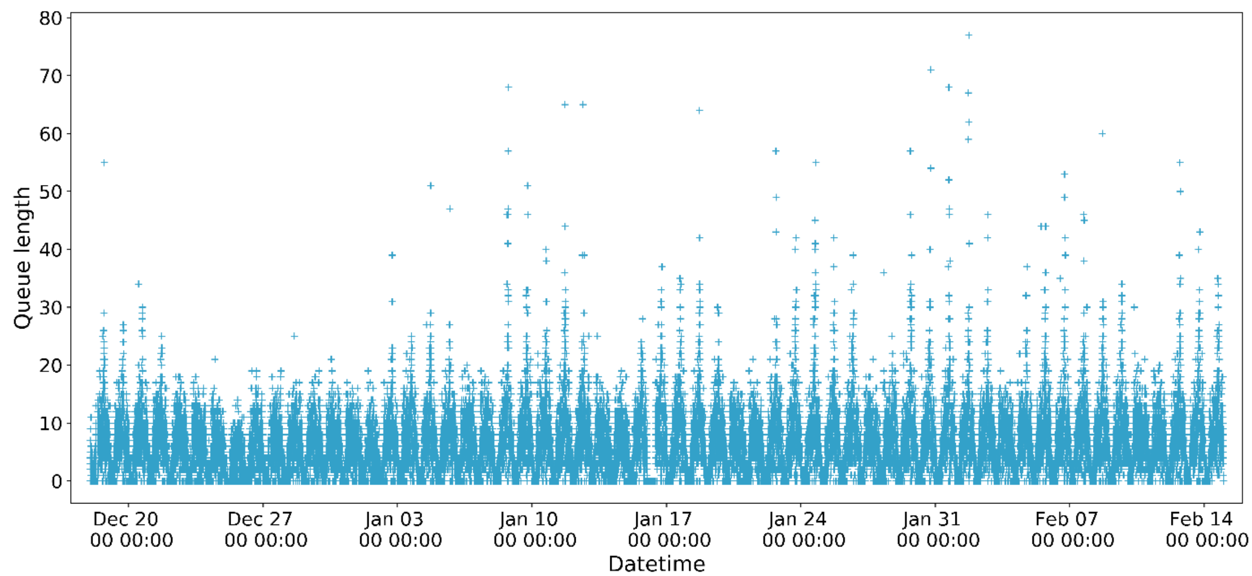
4.4.2 Data Preparation

In this study, we mainly focused on north through movements. We separated the data containing queue lengths (see Figure 4.3 (a)) and wait times for the north through movement. The data collected from the phase history log contain multiple queue lengths for a given direction (north through) for a single cycle period which means that the same queue (north through direction) was cleared multiple times within a single cycle period. For our study corridor, the cycle period usually varies in between 120sec to185sec.

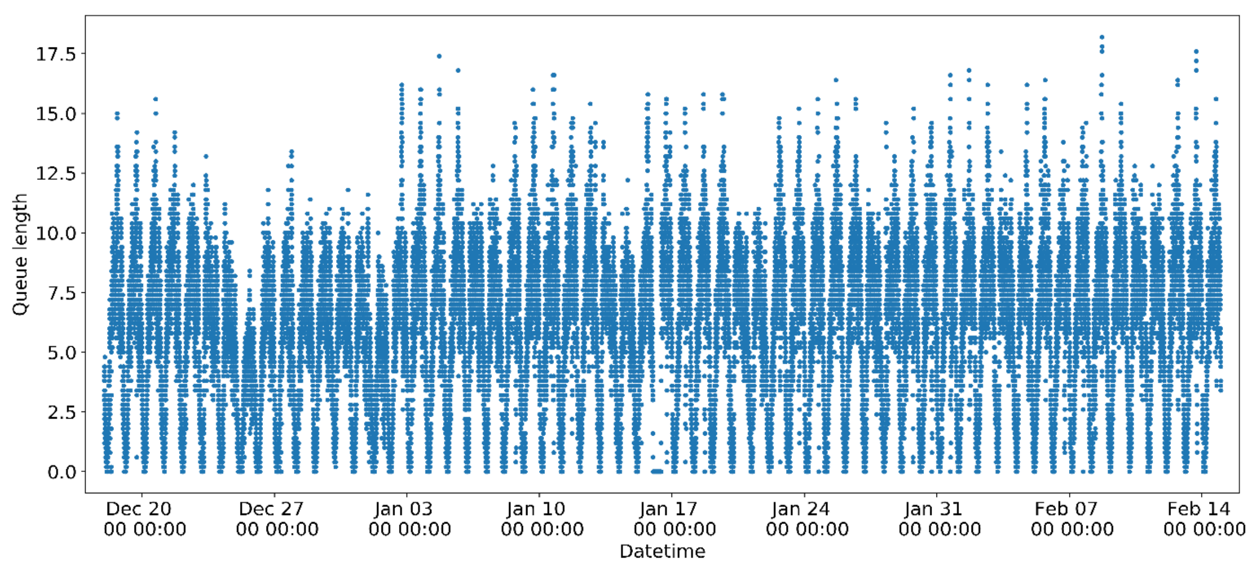
In general, the raw data collected from traffic sensors are subjected to errors. Several factors such as detector's malfunctioning, false encoding during storing into the server, bad weather conditions etc. can cause errors. To understand the quality of the data we plotted the queue length with respect to time. Figure 4.3 (a) shows that few data points drastically deviated from the regular trends indicating that the collected data contains a few outliers which might cause poor fitting of the model. Hence, we need to apply some data cleaning techniques to remove these outliers.

For cleaning the data, we applied two approaches. First, we considered the maximum possible queue length detection by the detectors. InSync Adaptive traffic controller depends on the mounted video cameras to detect the number of vehicles and how long the vehicles have been waiting. In some cases, the detection system is fused with loop detectors to assist the queue detection. The detectors are placed at a certain distance from the stop line at the upstream of the intersection. The distance varies between 285 feet and 484 feet (Traffic and Manual, 2016). Hence, maximum possible queue length detection by the detectors should be less than 35 (average vehicle length 14.5 feet). Considering this issue, we discarded the queue lengths greater than 40 from our analysis.

Then we used interquartile range to remove the outliers. We chose a boundary in between 1.5 times the interquartile range and remove the queue lengths which fall outside this boundary. For prediction purpose, we chose the cycle length as 120 sec and aggregated all the small queue lengths within a single cycle period. The objective is to predict the queue length for the next cycle (after 120 sec). Finally, we applied a rolling average method over a window size of 5 to reduce the noise. Figure 4.3(b) demonstrate the trends in queue length over time after cleaning.



(a) Queue length with outliers



(b) Queue length without outliers

Figure 4.3: Queue Length Variation over time for Alafaya Mcculloch Intersection

4.4.3 Experiment Results

To predict the queue lengths for the next cycle time, we trained the LSTM model with InSync data. We divided the data into two sets, first 80% of the data was used for testing and the next 20% of the data was used for validation. Finally, we trained the model to learn the pattern. For selecting the hyperparameter for the deep LSTM NN model we ran the SMBO algorithm with a predefined prior distribution of each parameter (Table 4.1) on different datasets corresponding to different intersections (1 to 9). Finally, we obtain the optimal combination of hyperparameters which works best for each dataset (Table 4.2).

Table 4.1: Prior Distribution of Each Parameter for Queue Length Prediction

Parameter Name	Distribution	Values
Number of Hidden Layers	Categorical	$x \in \{1,2\}$
Activation Function in each layer	Categorical	$x \in \{relu, tanh, sigmoid\}$
Number of Units in First Layer	Categorical	$x \in \{64,128,256,512\}$
Number of Units in Second Layer	Categorical	$x \in \{64,128,256,512\}$
Dropout in each layer	Uniform	$x \in [0,1]$
Optimizer	Categorical	$x \in \{adam, sgd, adagrad, rmsprop\}$
Batch Size	Categorical	$x \in \{360,720,1440\}$

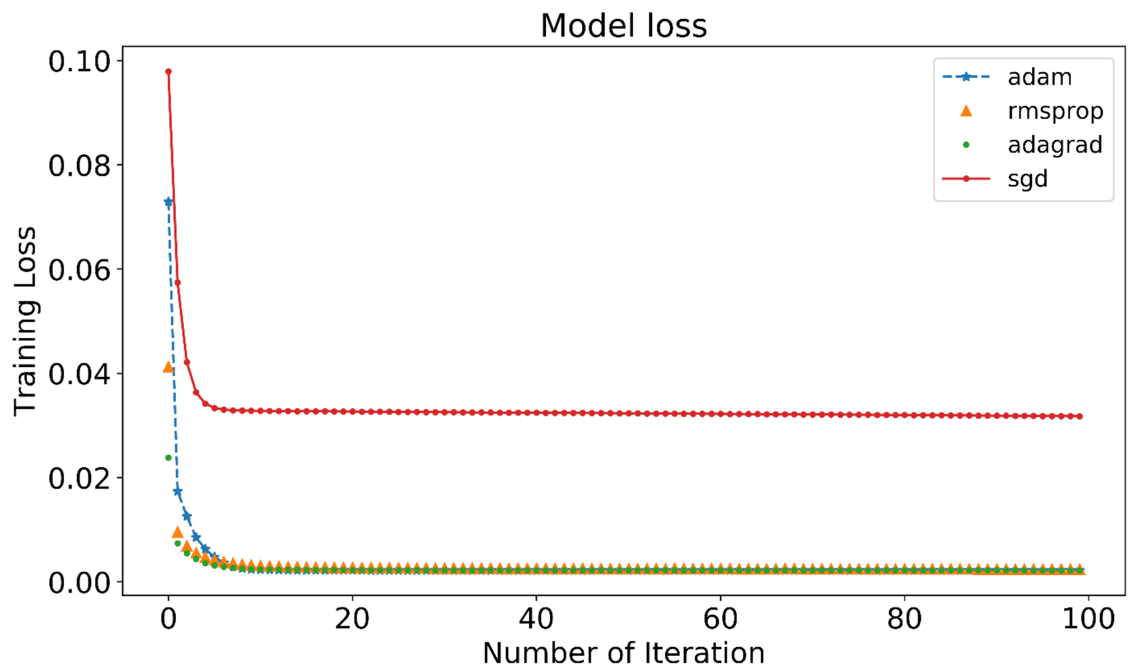


Figure 4.4: Variation of Training Loss per Iteration for Different Optimizer (Batch Size =1440)

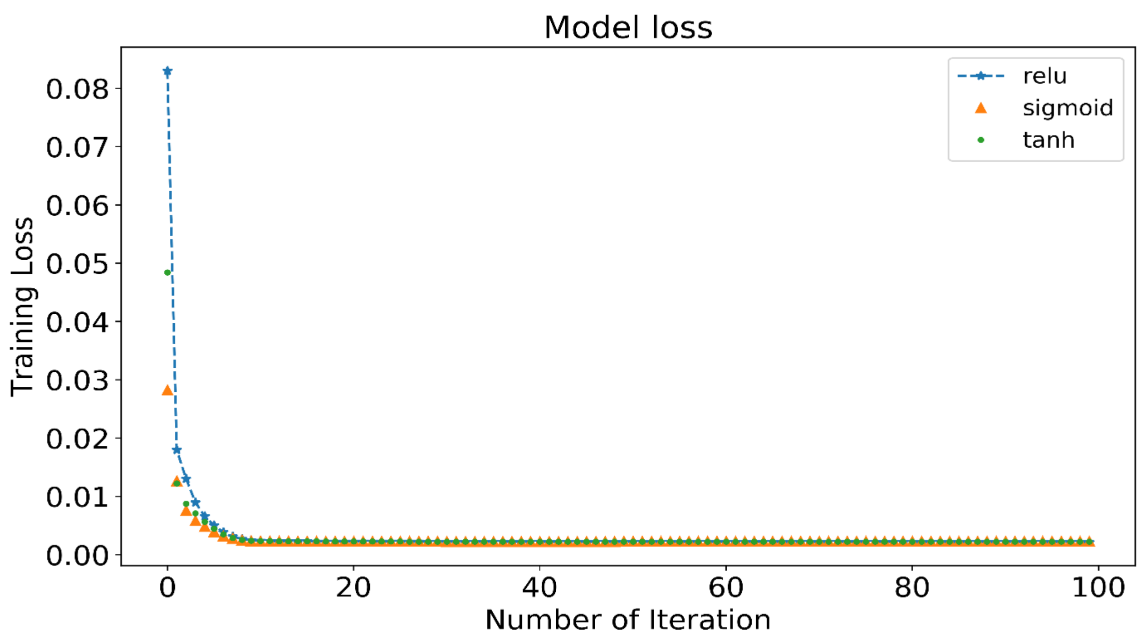


Figure 4.5: Variation of Training Loss per Iteration for Different Activation Function (Batch Size =1440)

From the optimization result we found that adam optimizer works better than rmsprop, adagrad and sgd optimizer. However as shown in figure 4.4, adam, rmsprop and adagrad have similar efficiency but adam optimizer converge faster than others. Hence, it takes less time to train the model. Figure 4.5 shows the training loss for different activation function. Both relu and tanh activation function work better, but if we choose sigmoid function the model starts overfitting at certain points before converging to the validation loss. Figure 4.6 shows the training and validation loss for the best model. We can see that the model converges after 70 iterations (epoch).

Table 4.2: Hyperparameters for best Performing Model for Queue Prediction

Number of Hidden Layers	Number of Hidden Units	Dropout	Activation Function	Optimizer
First	256	0.01655	relu	Adam
Second	128	0.00377	relu	

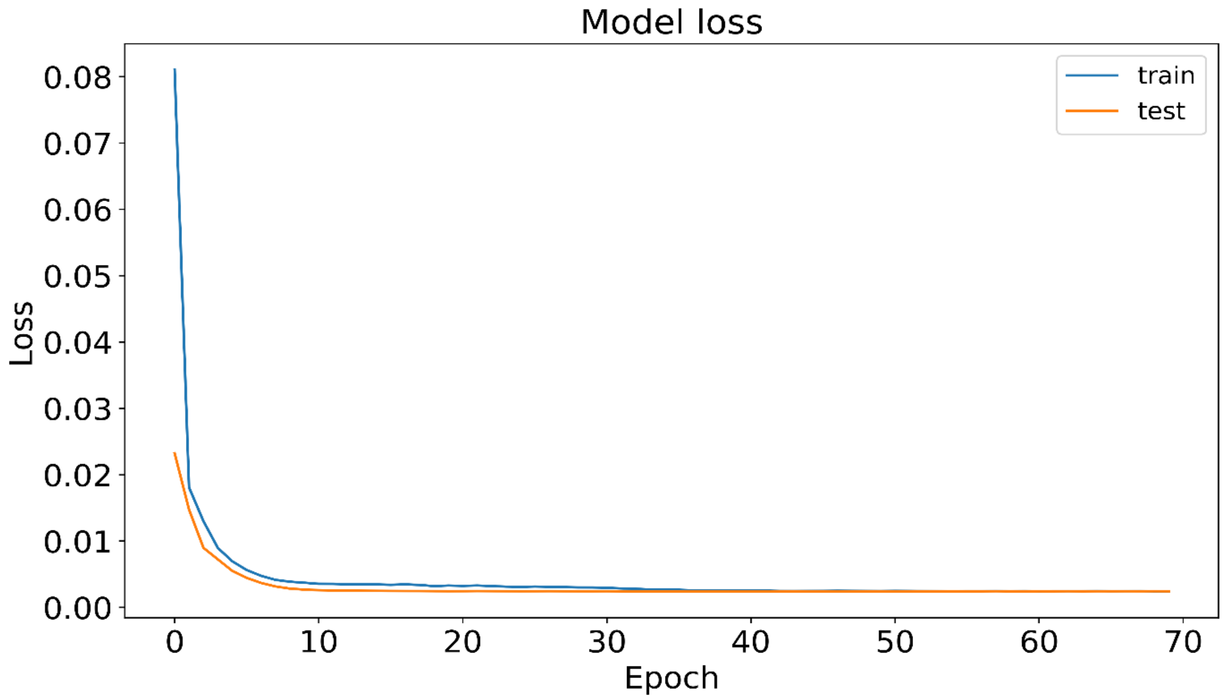


Figure 4.6: Training and Validation Loss for the Optimized Model

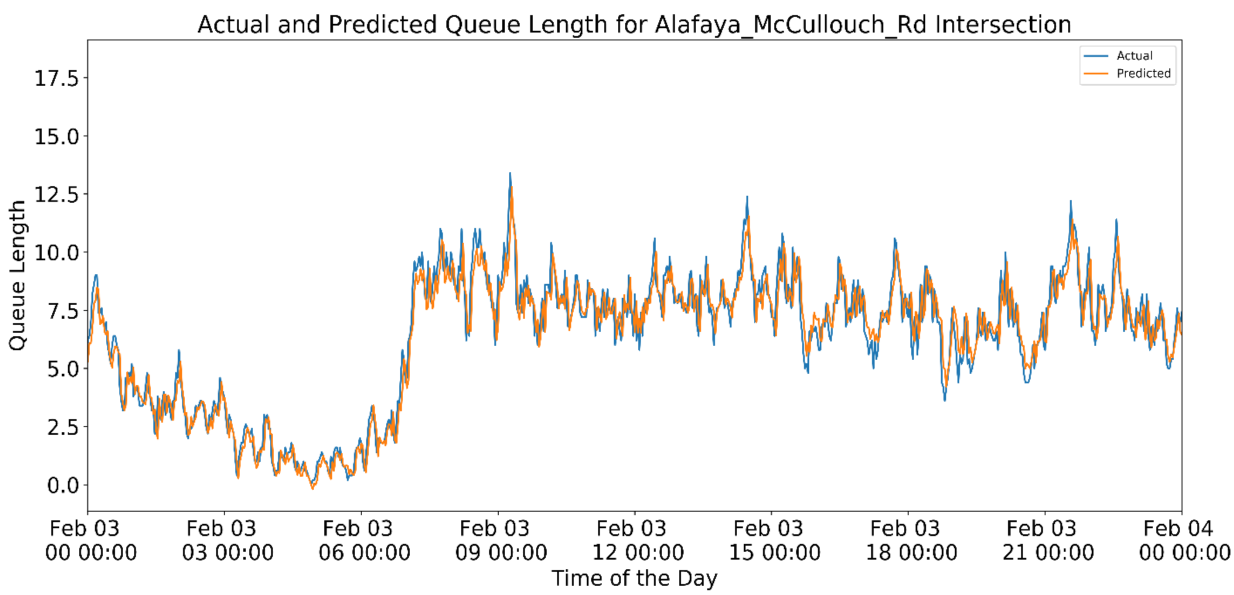


Figure 4.7: Actual and Predicted Queue Length for Alafaya and McCullouch Road Intersection (February 03, 2018)

As shown in Figure 4.7, the trained LSTM NN model performs very well to capture the variations of queue length over time. The difference between actual and predicted queue length is quite low. From Figure 4.8, we can observe that in maximum cases the difference between the actual and predicted value for different intersection varies from 0.3 to 1.2. We have calculated Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) as performance measures to check the accuracy of the implemented model. Performance metrics are defined as,

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}{n}} \qquad (7)$$

$$\mathrm{MAE} = \frac{\sum_{t=1}^{n}|y_t - \hat{y}_t|}{n} \qquad (8)$$

Figure 4.9 shows that in most cases the RMSE values are less than 1. The maximum RMSE value was found for Alafaya Trail and Corporate Blvd intersection. While for each intersection, the MAE value is less than 1 as well.

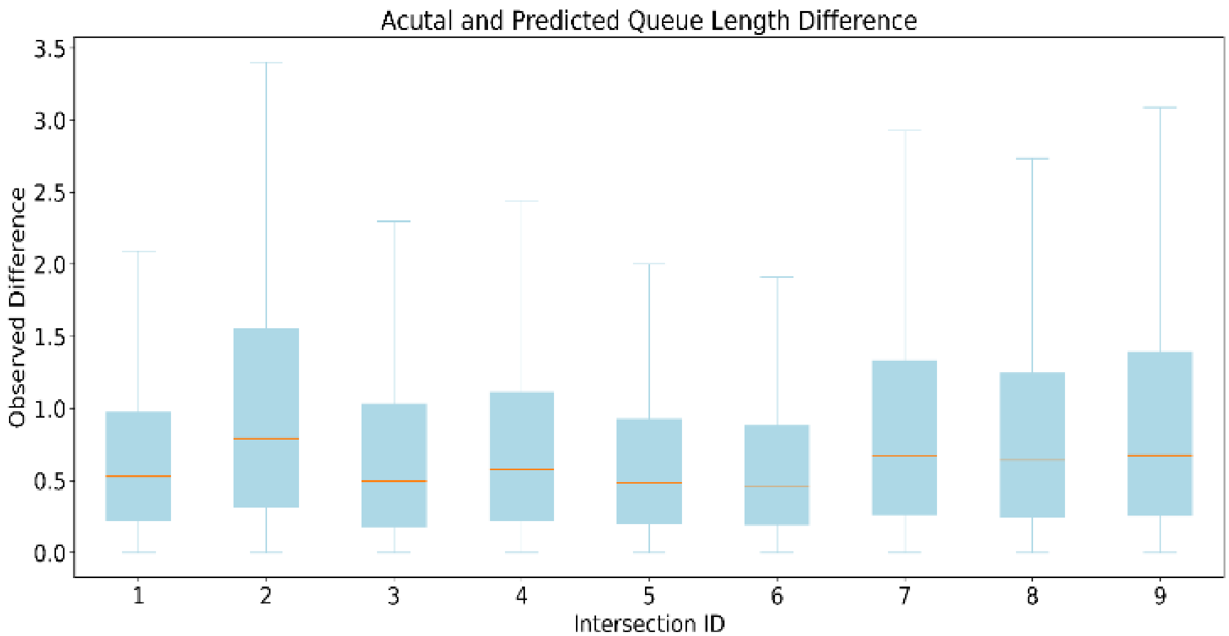


Figure 4.8: Distribution of the Difference between Actual and Predicted Queue Length

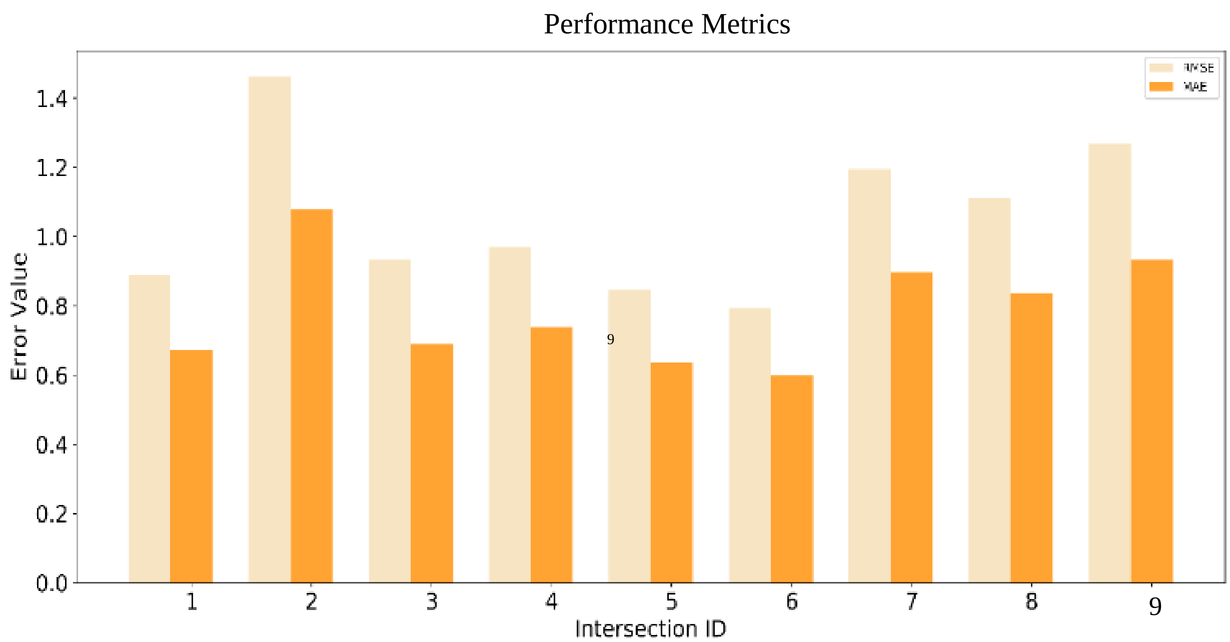


Figure 4.9: Variation of Performance Metrics for Different Intersections

4.5 Discussion

Queue length is one of the major performance measures to evaluate the performance of a traffic signal. In Advanced Traffic Control Systems, queue lengths have been used to optimize signal control parameters. In this study, we have developed a data-driven method to predict queue lengths in the next cycle from real-time traffic data. Assuming a connected corridor, we have implemented a deep LSTM-NN model to predict the queue length for the next cycle. Our deep learning method can capture the time-dependent patterns of traffic signal queues very well.

One of the major benefits of the proposed deep learning model is that it can be implemented in real time and can be updated based on real-time signal data. Moreover, it will reduce the dependency of the ATCS technologies on multiple detectors (e.g. loop detectors), hence reducing the overall maintenance cost to operate a system.

CHAPTER 5: CONCLUSIONS AND RECOMMENDATIONS

5.1 Summary

With the availability of multiresolution traffic data, deep leaning has created a unique opportunity to solve more complex traffic prediction problems. In this study, we developed data-driven solution to deal with two different problems using Deep LSTM NN model. But both problems have a similar goal, to predict traffic state (speed, signal queue length).

In the first problem, we develop a framework to predict the traffic speed for Interstate 75 (I-75) considering spatial and temporal dependency of the traffic state. We consider a connected corridor where the future traffic speed of the target link will depend on the current traffic state of the upstream, target and downstream link. To test the reliability of the model, we applied it to predict the traffic state during hurricane evacuation when traffic flow pattern shows irregular behavior. Our experiment result shows that our proposed modeling framework worked better in both regular and irregular traffic demand condition. Though LSTM NN model performed better than the traditional models, the accuracy of those models was reasonably good. Which means temporal and spatial dependency is critical in traffic state prediction and our proposed framework can capture this relation.

In our second problem, we consider a connected corridor of intersections where consecutive intersections will share information with each other and gather information of upcoming vehicles. We develop a data-driven approach to predict the lane-based queue length for an intersection. We anticipate that with emerging connected vehicles technologies and road environments, information (traffic state, queue length etc.) from one intersection will be easily available to another intersection. For our experiments, we use InSync Adaptive Signal data which provides queue lengths and wait times (time required for the first vehicle to clear the intersection) for different vehicular movements. We trained the LSTM-NN model to predict the queue length for the next cycle based on queue length and wait time of three consecutive intersections at the current cycle. Though we run this experiment to predict queue lengths for north through traffic, the Same methodology can be applied to predict queen lengths for other movements as well. Based on the accuracy metrics obtained from the experiment result we can conclude that LSTM NN performed well to predict the lane-based signal queue length.

One of the major benefits of data-driven solution method is that it can be applied in real time and can be updated using real-time data. However, the most critical issue with this data-driven method is the prediction accuracy. Since the traffic flow pattern follows a complex dynamic, it is difficult to capture those nonlinear patterns using traditional models. But deep leaning with layered nonlinear functions has the ability to capture high dimensional data representation which made it easier for us to deal with these complex problems. Hence, in the future with the introduction of connectivity (vehicle to vehicle and vehicle to infrastructure) these methods can be utilized to get the insights on future traffic. Especially during an emergency situation such as hurricane evacuation. Accurate traffic state prediction can largely improve the evacuation management system through proactive decision making. The findings of this study give evidence on the feasibility of this deep learning method to deal with traffic operation related problems.

5.2 Limitations and Future Research Direction

In our first problem, we developed a framework using LSTM NN model to predict the traffic speed. we choose only for four links of I-75, it should be tested using more links at a network level including other highways and arterial roads. More features traffic volume, delay, weather condition, etc. can be added from multiple data source using data fusion techniques to check whether such variables improve the performance of the model. The developed methodology can be implemented for predicting other traffic states such as travel time and traffic flow.

In our second problem, We develop a data-driven approach to predict the queue length for an intersection. we predicted the signal queue length only for through movements using historical queue length and wait time for through movements as input features, we can add more features related to vehicular traffic states (traffic flow, average travel time or speed) merging data from multiple sources to provide a more complete picture of signal states for better prediction. In our future study, we will do an experiment for a complete intersection considering the queue length for each lane. We will develop a data-driven optimization technique for the adaptive traffic control system based on the predicted queue lengths. Although we used a fixed cycle time but to implement the model in a practical field, we have to make the model more flexible so that it can predict the queue length for variable cycle time. Furthermore, we have to incorporate an algorithm that can update the next cycle time based on current traffic state and delay.

REFERENCES

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. https://doi.org/10.1038/nn.3331

Ahn, J., 2016. Highway traffic flow prediction using support vector regression and Bayesian classifier. 2016 Int. Conf. Big Data Smart Comput. 239–244. https://doi.org/10.1109/BIGCOMP.2016.7425919

An, C., Wu, Y.J., Xia, J., Huang, W., 2017. Real-time queue length estimation using event-based advance detector data. J. Intell. Transp. Syst. Technol. Planning, Oper. 0, 1–14. https://doi.org/10.1080/15472450.2017.1299011

Balke, K.N., Charara, H., Parker, R., 2005. Development of a Traffic Signal Performance Measurement System (TSPMS) 7, 83.

Ban, X. (Jeff), Herring, R., Hao, P., Bayen, A.M., 2010. Delay Pattern Estimation for Signalized Intersections Using Sampled Travel Times. Transp. Res. Rec. J. Transp. Res. Board 2130, 109–119. https://doi.org/10.3141/2130-14

Bergstra, J., Bardenet, R., Bengio, Y., Kégl, B., 2011. Algorithms for Hyper-Parameter Optimization. Adv. Neural Inf. Process. Syst. 2546–2554. https://doi.org/2012arXiv1206.2944S

Bergstra, J., Yamins, D., Cox, D.D., 2013. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. 12th PYTHON Sci. CONF. (SCIPY 2013) 13–20. https://doi.org/10.1088/1749-4699/8/1/014008

Billings, D., Jiann-Shiou, Y., 2006. Application of the ARIMA Models to Urban Roadway Travel Time Prediction-A Case Study. Systems, Man and Cybernetics, 2006. SMC'06. IEEE Int.

Conf. 2529–2534.

Cai, P., Wang, Y., Lu, G., Chen, P., Ding, C., Sun, J., 2016. A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transp. Res. Part C Emerg. Technol. 62, 21–34. https://doi.org/10.1016/j.trc.2015.11.002

Chang, G.L., Su, C.C., 1995. Predicting intersection queue with neural network models. Transp. Res. Part C 3, 175–191. https://doi.org/10.1016/0968-090X(95)00005-4

Chang, T.H., Lin, J.T., 2000. Optimal signal timing for an oversaturated intersection. Transp. Res. Part B Methodol. 34, 471–491. https://doi.org/10.1016/S0191-2615(99)00034-X

Chu, L., Oh, J.-S., Recker, W., 2005. Adaptive Kalman Filter Based Freeway Travel time Estimation. Transp. Res. Board 2005 Annu. Meet. 1–21.

Comert, G., 2013. Simple analytical models for estimating the queue lengths from probe vehicles at traffic signals. Transp. Res. Part B Methodol. 55, 59–74. https://doi.org/10.1016/j.trb.2013.05.001

Cui, Z., Wang, Y., 2017. Deep Stacked Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction 22–25.

Dave, M., n.d. The crucial reason Houston officials didn’t order evacuations before Harvey made landfall.

Deshpande, M., Bajaj, P.R., 2016. Performance analysis of support vector machine for traffic flow prediction. 2016 Int. Conf. Glob. Trends Signal Process. Inf. Comput. Commun. 126–129. https://doi.org/10.1109/ICGTSPICC.2016.7955283

FDOT, 2016. TSM&O Program: Advanced Signal Control Technology Guidelines 0450.

Feng, Y., Head, K.L., Khoshmagham, S., Zamanipour, M., 2015. A real-time adaptive signal control in a connected vehicle environment. Transp. Res. Part C Emerg. Technol. 55, 460–473. https://doi.org/10.1016/j.trc.2015.01.007

Geron, A., 2017. Hands-On Machine Learing With Scikit-Learn & Tensor Flow, Hands-on Machine Learning with Scikit-Learn and TensorFlow. O’Reilly Media.

https://doi.org/10.3389/fninf.2014.00014

Gers, F.A., Cummins, F., 1999. 1 Introduction 2 Standard LSTM 1–19.

Habtemichael, F.G., Cetin, M., 2016. Short-term traffic flow rate forecasting based on identifying similar traffic patterns. Transp. Res. Part C Emerg. Technol. 66, 61–78. https://doi.org/10.1016/j.trc.2015.08.017

Hao, P., Ban, X., 2015. Long queue estimation for signalized intersections using mobile data. Transp. Res. Part B Methodol. 82, 54–73. https://doi.org/10.1016/j.trb.2015.10.002

Hochreiter, S., Urgen Schmidhuber, J., 1997. Long Short-Term Memory. Neural Comput. 9, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hofleitner, A., Herring, R., Abbeel, P., Bayen, A., 2012. Learning the dynamics of arterial traffic from probe data using a dynamic bayesian network 13, 1679–1693. https://doi.org/10.1109/TITS.2012.2200474

Hutter, F., Hoos, H.H., Leyton-Brown, K., 2011. Sequential model-based optimization for general algorithm configuration. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) 6683 LNCS, 507–523. https://doi.org/10.1007/978-3-642-25566-3_40

Innamaa, S., 2005. Short-term prediction of travel time using neural networks on an interurban highway. Transportation (Amst). 32, 649–669. https://doi.org/10.1007/s11116-005-0219-y

Jeff Ban, X., Hao, P., Sun, Z., 2011. Real time queue length estimation for signalized intersections using travel times from mobile sensors. Transp. Res. Part C Emerg. Technol. 19, 1133–1156. https://doi.org/10.1016/j.trc.2011.01.002

Lecun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Lee, Y.L.Y., 2009. Freeway travel time forecast using artifical neural networks with cluster method. 2009 12th Int. Conf. Inf. Fusion 1331–1338.

Lighthill, M.J., Whitham, G.B., 1955. On Kinematic Waves. II. A Theory of Traffic Flow on Long

Crowded Roads. Proc. R. Soc. A Math. Phys. Eng. Sci. 229, 317–345. https://doi.org/10.1098/rspa.1955.0089

Liu, H.X., Wu, X., Ma, W., Hu, H., 2009. Real-time queue length estimation for congested signalized intersections. Transp. Res. Part C Emerg. Technol. 17, 412–427. https://doi.org/10.1016/j.trc.2009.02.003

Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C Emerg. Technol. 54, 187–197. https://doi.org/10.1016/j.trc.2015.03.014

May, A.D., 1975. Traffic Flow Theory- the Traffic Engineers Challenge.

Meng, M., Shao, C., Wong, Y., Wang, B., Li, H., 2015. A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. J. Cent. South Univ. 22, 779–786. https://doi.org/10.1007/s11771-015-2582-y

Mirchandani, P.B., Zou, N., 2007. Queuing Models for Analysis of Traffic Adaptive Signal Control. IEEE Trans. Intell. Transp. Syst. 8, 50–59. https://doi.org/10.1109/TITS.2006.888619

Myung, J., Kim, D.-K., Kho, S.-Y., Park, C.-H., 2011. Travel Time Prediction Using *k* Nearest Neighbor Method with Combined Data from Vehicle Detector System and Automatic Toll Collection System. Transp. Res. Rec. J. Transp. Res. Board 2256, 51–59. https://doi.org/10.3141/2256-07

Newell, G.F., 1965. Approximation Methods for Queues with Application to the Fixed-Cycle traffic Light. Soc. Ind. Appl. Math. 7, 223–240.

Oh, S., Byon, Y.J., Jang, K., Yeo, H., 2017. Short-term travel-time prediction on highway: A review on model-based approach. KSCE J. Civ. Eng. 1–13. https://doi.org/10.1007/s12205-017-0535-8

Park, D., Rilett, L.R., Han, G., 1999. Spectral Basis Neural Networks for Real-Time Travel Time Forecasting. J. Transp. Eng. 125, 515–523. https://doi.org/10.1061/(ASCE)0733-947X(1999)125:6(515)

Pel, A.J., Bliemer, M.C.J., Hoogendoorn, S.P., 2012. A review on travel behaviour modelling in dynamic traffic simulation models for evacuations. Transportation (Amst). 39, 97–123. https://doi.org/10.1007/s11116-011-9320-6

Polson, N., Sokolov, V., 2016. Deep Learning for Short-Term Traffic Flow Prediction 1–29. https://doi.org/10.1016/j.trc.2017.02.024

Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. Part C Emerg. Technol. 79, 1–17. https://doi.org/10.1016/j.trc.2017.02.024

Qiao, W., Haghani, A., Hamedi, M., 2013. A Nonparametric Model for Short-Term Travel Time Prediction Using Bluetooth Data. J. Intell. Transp. Syst. 17, 165–175. https://doi.org/10.1080/15472450.2012.748555

Rahman, R., Hasan, S., 2019. Real-time Signal Queue Length Prediction Using Long Short-Term Memory Neural Network, in: Transportation Research Board 98th Annual MeetingTransportation Research Board.

Rahman, R., Hasan, S., 2018. Short-Term Traffic Speed Prediction for Freeways During Hurricane Evacuation : A Deep Learning Approach 1291–1296. https://doi.org/10.1109/ITSC.2018.8569443

Richards, P.I., 1956. Shock Waves on the Highway. Oper. Res. 4, 42–51. https://doi.org/10.1287/opre.4.1.42

Robertson, D., 1969. TRANSYT: A Traffic Network Study Tool, Road Research Laboratory.

Roy, K., 2018. Understanding Crisis Communication and Mobility Resilience during Disasters from Social Media STARS Citation.

Roy, K.C., Hasan, S., 2019. Modeling the Dynamics of Hurricane Evacuation Decisions from Real-time Twitter Data.

Roy, K.C., Hasan, S., 2018. Quantifying Human Mobility Resilience to Extreme Events Using Geo-located Social Media Data, in: Transportation Research Board 97th Annual Meeting.

Schrank., D., Eisele., B., Lomax., T., Bak., J., 2015. 2015 Urban Mobility Scorecard. Texas A&M

Transp. Institue 39, 5. https://doi.org/DTRT06-G-0044

Seo, T., Bayen, A.M., Kusakabe, T., Asakura, Y., 2017. Traffic state estimation on highway: A comprehensive survey. Annu. Rev. Control 43, 128–151. https://doi.org/10.1016/j.arcontrol.2017.03.005

Sharma, A., Bullock, D.M., Bonneson, J.A., Sharma, A., Bullock, D.M., Bonneson, J.A., 2007. Input-Output and Hybrid Techniques for Real- Time Prediction of Delay and Maximum Queue Length at Signalized Intersections Delay and Maximum Queue Length at Signalized Intersections. Transp. Res. Rec. J. Transp. Res. Board 2035, 69–80. https://doi.org/10.3141/2035-08.

Smaglik, E., Sharma, A., Bullock, D., Sturdevant, J., Duncan, G., 2007. Event-Based Data Collection for Generating Actuated Controller Performance Measures. Transp. Res. Rec. J. Transp. Res. Board 2035, 97–106. https://doi.org/10.3141/2035-11

Smith, S.F., Barlow, G.J., Xie, X.-F., Rubinstein, Z.B., 2013. Smart Urban Signal Networks: Initial Application of the SURTRAC Adaptive Traffic Signal Control System. Icaps 434–442.

Stephanopoulos, G., Michalopoulos, P.G., Stephanopoulos, G., 1979. Modelling and analysis of traffic queue dynamics at signalized intersections. Transp. Res. Part A Gen. 13, 295–307. https://doi.org/10.1016/0191-2607(79)90028-1

Svozil, D., Kvasnicka, V., Pospichal, J., 1997. Introduction to multi-layer feed-forward neural networks. Chemom. Intell. Lab. Syst. 39, 43–62. https://doi.org/10.1016/S0169-7439(97)00061-0

Tiaprasert, K., Zhang, Y., Wang, X.B., Zeng, X., 2015. Queue Length Estimation Using Connected Vehicle Technology for Adaptive Signal Control. IEEE Trans. Intell. Transp. Syst. 16, 2129–2140. https://doi.org/10.1109/TITS.2015.2401007

Traffic, T., Manual, D., 2016. CHAPTER 8 TRAFFIC SIGNAL DESIGN – 1–18.

Vigos, G., Papageorgiou, M., Wang, Y., 2008. Real-time estimation of vehicle-count within signalized links. Transp. Res. Part C Emerg. Technol. 16, 18–35. https://doi.org/10.1016/j.trc.2007.06.002

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014a. Short-term traffic forecasting: Where we are and where we're going. Transp. Res. Part C Emerg. Technol. 43, 3–19. https://doi.org/10.1016/j.trc.2014.01.005

Vlahogianni, E.I., Karlaftis, M.G., Golias, J.C., 2014b. Short-term traffic forecasting: Where we are and where we're going. Transp. Res. Part C Emerg. Technol. 43, 3–19. https://doi.org/10.1016/j.trc.2014.01.005

Webster, F. V, 1957. Traffic signal settings. Road Res. Tech. Pap. 39.

Wu, C., Wei, C., Su, D., Chang, M., Ho, J., 2004. Travel time prediction with support vector regression. Proc. 2003 IEEE Int. Conf. Intell. Transp. Syst. 2, 1438–1442. https://doi.org/10.1109/ITSC.2003.1252721

Wu, C.H., Ho, J.M., Lee, D.T., 2004. Travel-time prediction with support vector regression. IEEE Trans. Intell. Transp. Syst. 5, 276–281. https://doi.org/10.1109/TITS.2004.837813

Xu, J., Rahmatizadeh, R., Turgut, D., 2017. Real-Time Prediction of Taxi Demand Using Recurrent Neural Networks 1–10.

Yanjie Duan, Yisheng Lv, Fei-Yue Wang, 2016. Travel time prediction with LSTM neural network. 2016 IEEE 19th Int. Conf. Intell. Transp. Syst. 1053–1058. https://doi.org/10.1109/ITSC.2016.7795686

Yu, B., Song, X., Guan, F., Yang, Z., Yao, B., 2016. k-Nearest Neighbor Model for Multiple-Time-Step Prediction of Short-Term Traffic Condition. J. Transp. Eng. 142, 04016018. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000816

Yu, J., Chang, G.-L., Ho, H.W., Liu, Y., 2008. Variation Based Online Travel Time Prediction Using Clustered Neural Networks. 2008 11th Int. IEEE Conf. Intell. Transp. Syst. 85–90. https://doi.org/10.1109/ITSC.2008.4732594