Final Report

Estimating Switching Times of Actuated Coordinated Traffic Signals: A Deep Learning Approach

Seifeldeen Eteifa
Virginia Tech Transportation Institute
3500 Transportation Research Plaza
Blacksburg, VA 24061
seteifa@vt.edu

Hesham A. Rakha, Ph.D., P.Eng.
Charles E. Via, Jr. Department of Civil and Environmental Engineering
Virginia Polytechnic Institute and State University
3500 Transportation Research Plaza, Blacksburg, VA 24061
Phone: (540) 231-1505 - Fax: (540) 231-1555
hrakha@vt.edu

Hoda M. Eldardiry, Ph.D.
Department of Computer Science
Virginia Polytechnic Institute and State University
3160D Torgersen Hall, Blacksburg, VA 24061
Phone: (540) 231- 0906
hdardiry@vt.edu

Date
November 2021

Prepared for the Urban Mobility & Equity Center, Morgan State University, CBEIS 327, 1700 E. Coldspring Lane, Baltimore, MD 21251

ACKNOWLEDGEMENT

This research was funded by the Urban Mobility & Equity Center at Morgan State University and the University Transportation Center(s) Program of the U.S. Department of Transportation.

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the U.S. Department of Transportation's University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

1. Report No. UMEC-039	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Estimating Switching Times of Actuated Coordinated Traffic Signals: A Deep Learning Approach		**5. Report Date** November 2021
		6. Performing Organization Code
7. Author(s) Seifeldeen Eteifa Hesham A. Rakha (ORCID # 0000-0002-5845-2929) Hoda M. Eldardiry		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Virginia Tech Transportation Institute 3500 Transportation Research Road Blacksburg, VA 24061		**10. Work Unit No.**
		11. Contract or Grant No. 69A43551747123
12. Sponsoring Agency Name and Address US Department of Transportation Office of the Secretary-Research UTC Program, RDT-30 1200 New Jersey Ave., SE Washington, DC 20590		**13. Type of Report and Period Covered** Final Oct 2020- October 2021
		14. Sponsoring Agency Code
15. Supplementary Notes		

16. Abstract

Acceleration and deceleration maneuvers at signalized intersections are a major hindrance to fuel-efficient vehicle operations. Green Light Optimal Speed Advisory (GLOSA) allows for the control of vehicles in a fuel-efficient manner but requires reliable estimates of signal switching times. This study attempts to utilize data from actuated coordinated signalized intersections in Northern Virginia along with multiple deep learning and machine learning techniques to provide estimates of traffic signal switching times from green to red and vice versa. These estimates can be used to enable more fuel-efficient operation using GLOSA and eco-driving. They can also be used to mitigate dilemma zone safety concerns. A comparative analysis is conducted between the different techniques used and their pros and cons in terms of prediction errors and robustness to different traffic conditions.

17. Key Words: Deep learning, Actuated Signals, Green Light Optimal Speed Advisory, Eco-driving	**18. Distribution Statement**		
19. Security Classif. (of this report) : Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 46	**22. Price**

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

INTRODUCTION

The transportation industry is one of the largest industries in the U.S. in terms of its contribution to fossil fuel and energy consumption and pollution. In 2020, the transportation sector alone accounted for 26% of the total consumption of energy in the U.S. (*1*). Out of that energy consumption, more than 90% comes from petroleum, making the transportation industry one of the major polluting industries (*1*). In 2019, the transportation industry was considered the most polluting industry in terms of greenhouse gas emissions, contributing to global warming as shown in Figure 1. The greenhouse gas emissions of the transportation industry alone account for 29% of the total greenhouse gas emissions in the U.S., making transportation emissions a major contributor to global warming.

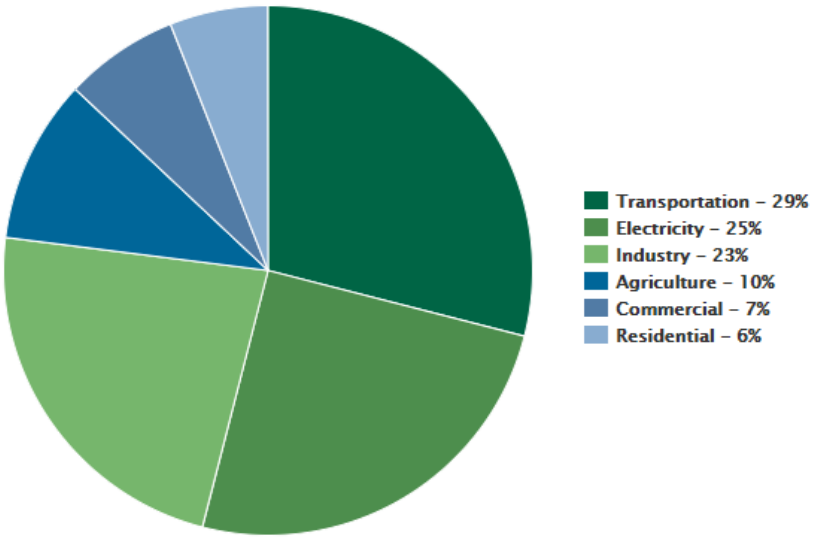

Figure 1: Contribution of different industries to Greenhouse Gas Emissions (Source: U.S. Environmental Protection Agency website)

Urban traffic emissions constitute a major problem for urban air quality (*2*). Within a city, traffic flow is continuously interrupted by red lights, resulting in numerous accelerations and decelerations at signalized intersections and significant stop and go movements. The constant acceleration and deceleration resulting from stop and go movements can increase fuel consumption and greenhouse gas emissions. Most cities depend on signalized intersections for their main arterials, making the stop and go movements a key problem affecting urban air quality as well as fuel consumption.

Stop and go movements at signalized intersections introduce additional problems to the

transportation system apart from fuel consumption and emissions. A major problem is safety concerns because a lack of knowledge about when a signal turns red catches drivers in what is known as yellow light dilemma zone (*3; 4*). Yellow light dilemma zones occur when the driver cannot make a clear-cut decision on whether to proceed through an intersection during the amber interval or brake forcefully to stop the vehicle before the stop line. An additional concern is the traffic throughput, where consistent stop and go can generate shockwaves that can reduce the overall traffic throughput in some cases (*5*). This introduces additional delays and increases the overall vehicle travel time in the system.

Green Light Optimal Speed Advisory and Eco-driving

Green Light Optimal Speed Advisory (GLOSA) and eco-driving have been proposed as potential ways to reduce stop and go waves. This can reduce fuel and energy consumption and emissions at signalized intersections by reducing unnecessary stops and adjusting vehicle speeds to allow vehicles to smoothly traverse their routes through green waves if possible.

Rakha and Kamalanathsharma (*6*) proved that by incorporating microscopic fuel consumption models, optimization of the fuel consumption profile of a vehicle approaching an intersection can be achieved with V2I (vehicle to infrastructure) communication about signal timing. The incorporation of the microscopic fuel consumption model yielded a better result than other simplified objective functions. Rakha et al. (*7*) created a velocity advisory tool to compute the fuel optimal velocity profile. The tool considers the information about signal change times, vehicle properties, communication from the lead vehicle and vehicle dynamics as well as a microscopic fuel consumption model as inputs. These inputs are used to run a complex optimization to come up with the optimal trajectory in terms of fuel efficiency. Dynamic programming was used along with path finding algorithms to create optimized upstream traffic trajectories while comparing various discretized cases for downstream traffic yielding a faster more efficient computational approach (*8*). The proposed approach yielded fuel savings of 32% and reduced travel time by 19%. It also provided potential benefits for following vehicles.

Chen et al. (*9*) conducted field testing of eco-driving systems on various grade levels, addressing issues such as communication latencies, computation time, data errors and driving comfort and achieving fuel savings amounting to between 8.4% and 17.4%. Almanaa et al. (*10*) conducted another field test study of eco-cooperative adaptive cruise control (Eco-CACC) and found that automated systems were superior to human control, leading to about a 19% reduction in fuel consumption when implementing Eco-CCC. The study also found that fuel savings of up

to 31% for downhill driving and 9% when going uphill can be attained through Eco-CACC.

Stahlman et al. (*11*) examined the existing GLOSA systems and introduced metrics to make the inputs to GLOSA simulation studies less optimistic and more realistic in terms of the communication latency and technology available in the field. Karoui et al. (*12*) add driver reaction time and GLOSA market penetration rate to existing simulation studies, yielding more realistic expectations in terms of fuel consumption and reduced vehicle stops. Yang et al. (*13*) developed an eco-cooperative adaptive cruise control (Eco-CACC) algorithm at signalized intersections, taking into account the queuing at the intersection, that led to saving up to 40% of fuel consumption. The study also concluded that to attain a reduction in fuel consumption, the market penetration rate of Eco-CACC has to be at least 30%. Suzuki and Marumo (*14*) used simulation models to prove GLOSA implementation can lead to improvements in traffic safety and fuel efficiency through decreasing deceleration rates in the vicinity of intersections.

All these studies agree that Eco-CACC, eco-driving or GLOSA are ways to reduce the fuel and energy consumption, safety concerns and in some cases travel time through signalized intersections. The studies also show that having a reliable estimate of the traffic signal switching times is crucial to enable GLOSA, eco-driving or Eco-CACC (*15*). The traffic signal switching time is, in fact, the key piece of information required for transmission from the infrastructure to the vehicle through V2I technology in order to be able to compute an optimal trajectory and attain the aforementioned benefits. This project aims to utilize machine learning to obtain a reasonable estimate of this traffic signal switching time by utilizing LSTM deep learning.

Uses of Machine Learning in Traffic Signal Systems Research

The use of machine learning (ML) with traffic signal systems involves two main tasks: traffic signal control and traffic signal prediction. The traffic signal control application is concerned with using data to change the controller settings to make it more adaptive to traffic conditions and improve system performance, for example, in terms of vehicle delay or emissions. Traffic signal prediction, on the other hand, is concerned with being able to predict the switching time of the actuated traffic signals under the existing settings and traffic conditions. This study is not focused on changing the controller settings, rather it is concerned with the prediction of the actuated traffic signal switching times with the controller settings currently implemented in the field. Providing a reliable estimate of these switching times is very practical, and communicating these estimates to vehicles can enable GLOSA to be more widely adopted.

Most of the existing body of knowledge in the literature utilizes ML combined with agent-based modelling for the traffic signal control task. Several studies used reinforcement learning to optimize the signal timings to adapt to traffic demand (*16-19*). Other studies utilized multi-agent

systems with machine learning or game theory to optimize systems of traffic signal controllers (*20-23*).

Only a few studies have tackled the idea of providing a prediction of actuated traffic signal switching times under existing controller settings to enable GLOSA. Support vector machines have been used as a simple classifier for the green time in a simple actuated traffic signal with only four different options for possible green times and were able to provide high accuracy classification predictions (*24*). Moreover, graph-based Bayesian methods have been used to predict the time until the next phase based on time of day, controller logic and detector data (*15*). This study expands on a previous study that used LSTM neural networks to predict the remaining time to change from green to red or vice versa for actuated traffic signals (*25*).

Problem Statement

Obtaining the traffic signal switching time for a fixed time signal is a very simple task. The switching times can be directly read from the signal timing plan and communicated to the vehicles through the V2I Signal Phase and Timing (SPaT) message stream. For the actuated traffic signals, on the other hand, this task is much more complex. Unlike fixed time traffic signals, actuated traffic signals have a minimum and maximum time for each phase. Every time a vehicle arrives, the phase time is extended by a preset amount known as passage time as long as it does not exceed the maximum time setting (Figure 2). This makes predicting how long each phase will be particularly tricky as the termination time of each phase depends not only on the number of vehicle arrivals but also on the pattern of arrivals, which can be random based on several studies (*26-28*).

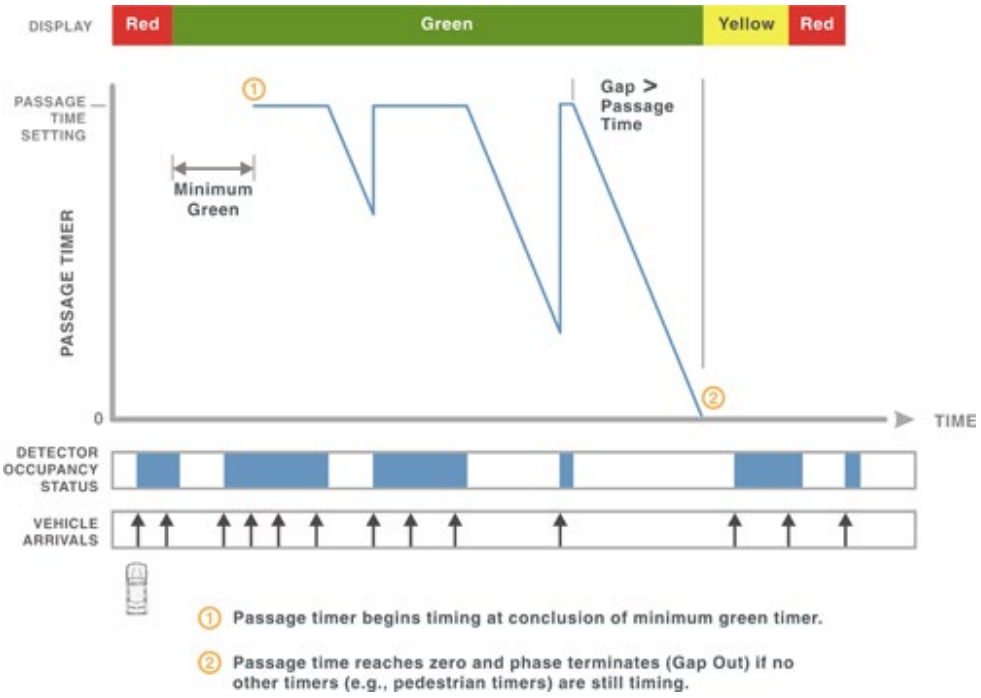

Figure 2: Actuated Traffic Signal Control (Source: Federal Highway Administration Signal Timing Manual)

Existing traffic signal controllers are very flexible in terms of accommodating different road users and allocating varying times to different movements. But this flexibility makes them less predictable because the more flexible the controller is to accommodate different traffic conditions, the less predictable its timing is. The complexity of predicting the traffic signal state can be attributed to two factors. The first is the controller logic where the D4 controller logic is highly adaptable and allows for different ways to adapt to the incoming traffic. Examples of this adaptability include options like having floating green times that can be allocated to different movements, allowing all settings to vary according to time of day, allowing locked calls for vehicles or pedestrians to be placed on a certain phase, and allowing reserving left turning vehicles that have not been serviced, meaning that one cycle can have two left turn phases, if needed. Even if these features are not used, they make the traffic signal highly unpredictable and increase the complexity of the prediction task. The second factor is the highly stochastic nature of traffic and pedestrian arrivals. Several studies have attempted to predict the effect of traffic arrivals on signal timing using low frequency probe vehicle data, GPS trajectory big data and data from upstream intersections combined with platoon dispersion modelling (*26-28*). Their results show the highly stochastic nature of traffic arrivals.

Therefore, in order to obtain the traffic signal switching times for actuated signals, there has to be some way of implicitly identifying the traffic arrivals and whether those arrivals would be so few that the minimum green would be sufficient, so many that the maximum green

time will be reached (max out), or somewhere in between where the time is between the minimum and maximum time (gap out). That knowledge is required to obtain a reasonable estimate of when the traffic signal is going to switch to enable eco-driving and GLOSA algorithms to control the vehicle velocity ahead of time to achieve optimal fuel consumption and reduce emissions. Ideally, the required means to predict the traffic signal switching time should take into account all traffic conditions and pedestrian traffic as well as signal timing parameters. It should also be scalable and replicable to allow for large-scale implementation within smart infrastructure.

To improve the model scalability and applicability, the model should be relatively easy to train within a reasonable amount of computational time and with a reasonable amount of data and computational resources including memory, GPU or CPU processing power and data storage. It should also be sufficiently robust to allow it to operate under varying conditions without much loss of accuracy in predicting the traffic signal switching times. This project works on not only providing the models but also improving their performance, robustness and reducing their computational complexity, memory and storage needs.

PROPOSED APPROACH

This research proposes the use of Long-short term memory (LSTM) recurrent neural networks as an approach for the prediction of the signal switching times. This approach allows for not only including all the data relevant to the prediction but also recognizes the temporal dependencies between the data elements at different time steps. This is allowed by the special building block of the LSTM network known as the LSTM cell that models temporal dependency among variables. This feature of capturing the temporal dependencies made LSTM lend itself to areas such as language modelling, speech recognition and stock market price prediction (*8-10*). LSTM networks have also been used in transportation applications. They have also been applied to areas with strong reliance on temporal trends such as predictions for roadway link travel time, traffic flow, and accident risk and severity (*29-33*).

The key idea is to recognize the importance of temporal dependencies among signal states at different times, as well as the temporally dependent nature of most data used in predicting the signal switching times. This includes traffic volumes, speed, traffic arrivals and pedestrian arrivals data. These parameters not only depend on the time of day but also can show distinct trends in the very short term. For example, if the traffic volume on one of the roads is increasing over the past few cycles, it might have a higher probability of increasing in the current cycle.

After building the baseline model, this project examines the effect of varying the time window in the past at which data can be fed to the LSTM neural network and the effect of that on

the prediction performance. The importance of choice of correct look-back periods for the model is twofold. First, it affects the overall model performance and reliability of predictions. Second, it affects the overall training time required for the model, thus minimizing the hardware requirements for field deployment of a model that can learn from live data and improve while providing predictions in the field. This provides a first step toward better understanding the effect of different time windows on the predictions of LSTM models in the context of actuated traffic signal switching times. It aims at demonstrating the sufficiency of the data to allow the model to make more reliable predictions.

Another aspect is assessing the model robustness and generalization performance. This allows our research to be a guideline for other researchers for the sufficiency of training data. It is also intended to provide some sense of how often trained models could be retrained to maintain acceptable prediction performance. In this project, different regularization methods are also assessed in terms of their contribution to both the prediction performance and the generalization performance.

This study could act as a baseline for other researchers working with similar data to estimate the amount of data they need to collect and time windows to work with. It can also act as a first step toward deploying deep learning methods within connected infrastructure. A simple example would be to modify the Signal Phasing and Timing (SPaT) message stream by adding the most likely signal switching time using the models presented in this paper. This can significantly improve the decision making of intelligent vehicles when using eco-driving systems, leading to more fuel-efficient vehicle operations.

Objectives and Contribution

This research has seven main objectives:

- Outline a detailed practical framework for the use of LSTM in traffic signal switching time prediction.
- Test the framework on signal switching time prediction using field data.
- Compare the effect of using different loss functions on the training process.
- Propose a novel loss function that enables the predictive model to generalize well across various time horizons.
- Identify the effect of the look-back period on short- and long-term traffic signal state prediction.
- Assess the performance of the LSTM models trained prior to the COVID-19 pandemic on data gathered during the pandemic.
- Test the effect of different regularization parameters on the overall performance of the LSTM-based models.

To the best of our knowledge, this research is the first to implement LSTM for actuated coordinated traffic signal switching time prediction. Unlike most studies that either focus on time to green (TTG) switching time or time to red (TTR) switching time, our proposed approach utilizes a single model for predicting both TTG and TTR switching times. This model provides a holistic approach which considers all the different users of the transportation network including the effect of pedestrian traffic. The effect of pedestrian traffic has not yet been included in any of the existing models for predicting actuated signal switching times. This can be mainly attributed to the fact that data regarding pedestrian actuations from push buttons and pedestrian phase timing and exit modes were not readily available. Finally, the modelling approach described in this report is highly data-driven and can be extended from its current reliance on historical data to a live implementation that continuously uses data to improve predictions.

The remainder of the paper is divided into three main parts followed by the conclusion. Part 1 describes the methodology and results for building the baseline models and assessing the performance of different loss functions and introducing a new proposed loss function. Part 2 covers optimizing the model look-back time window and looking at the effect for short- and long-term prediction performance. Part 3 covers comparing different model variants based on different regularization parameters and assessing the overall robustness of the models in terms of prediction performance after the COVID-19 pandemic.

PART 1: BASE MODEL DEVELOPEMENT AND LOSS FUNCTION COMPARISON

Methodology

To be able to provide predictions for the traffic signal switching times, a four-step research methodology was undertaken. The first step was data collection which entailed gathering the data broadcasted on the website every second and saving it to the database. The second step was data validation and preparation which included formatting the data to generate 120 second sequences of data to be used by the LSTM network for prediction. This involved handling missing data due to connection losses and server downtime as well as extracting the relevant data features to enhance the prediction. The third step was designing, coding, and tuning the LSTM neural networks to improve the prediction accuracy using cross validation. The final step was validating and testing the trained network on out of sample data.

Data Description

Data was collected from Gallows Road, which is part of state route 650 in Fairfax County in

Northern Virginia. Gallows Road is a major commuter road connecting Tysons Corner, a census-designated place which is a major shopping destination in Fairfax County, and Annandale, a census-designated place with multiple residential and commercial areas. The roadway has a high traffic volume; the average daily traffic is 39,000 users and the average annual weekday traffic is 42,000 users. The land use of areas surrounding Gallows Road is mostly commercial despite the area having a few residential developments.

Data gathering scripts have been deployed to obtain data from the intersection between Gallows Road and Gatehouse Drive (Figure 3). A database of historical data was created including 83 days of data for each of the intersections. The obtained data spans the period from July 2019 until December 2019. The data was obtained from the Virginia Smarterroads online portal. The online portal broadcasts data every one second, which is very descriptive of the traffic state.

Key Data Elements

The Smarterroads data are highly detailed in terms of coverage of the traffic conditions. Data are provided every second and are very fine grained. Data includes many elements pertaining to timing, controller settings, vehicles and pedestrians as shown in Table 1. Some of the advanced elements include the signal status, and whether it is in its minimum green time, in passage time allowing more time based on actuations, or terminating. Another advanced data element is the termination method, whether the controller reaches the maximum green (max out) or did not receive any actuations within the passage time (gap out).

Table 1 Data Elements

Data Elements			
Signal Timing	**Controller Settings**	**Vehicles**	**Pedestrians**
• Time of Day • Duration signal spent in current phase • Status (Minimum green/ Passage time/ Terminating)	• Cycle Length • Offset from Upstream signal • Timing Plan ID	• Current Phase • Speed • Volume • Occupancy • Actuations • Exit Mode (Max out/ Gap out)	• Current Phase • Push Button Status

Figure 3: Intersection layout

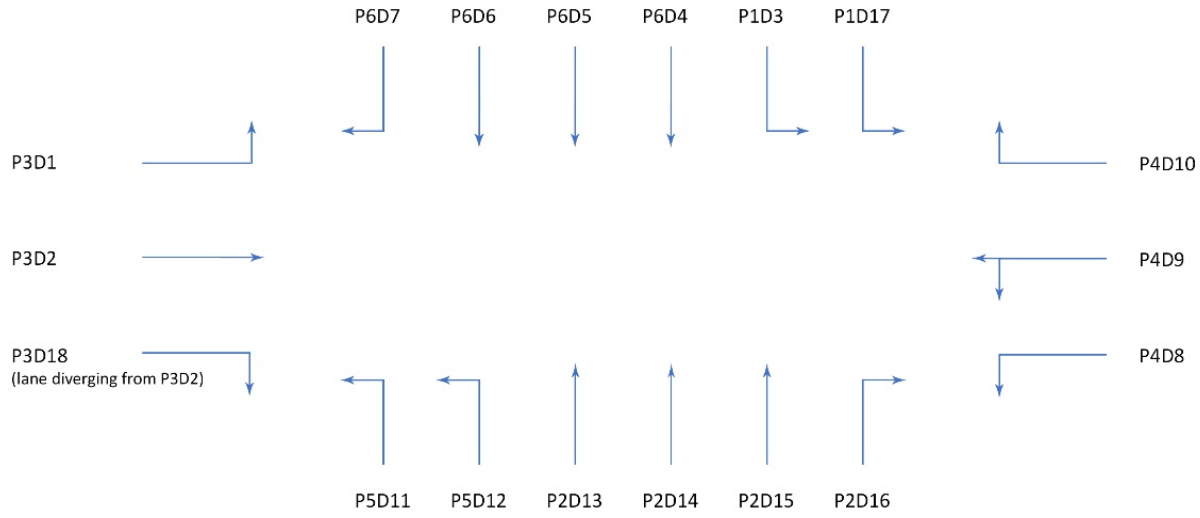

Figure 4: Lane configuration and detector placement

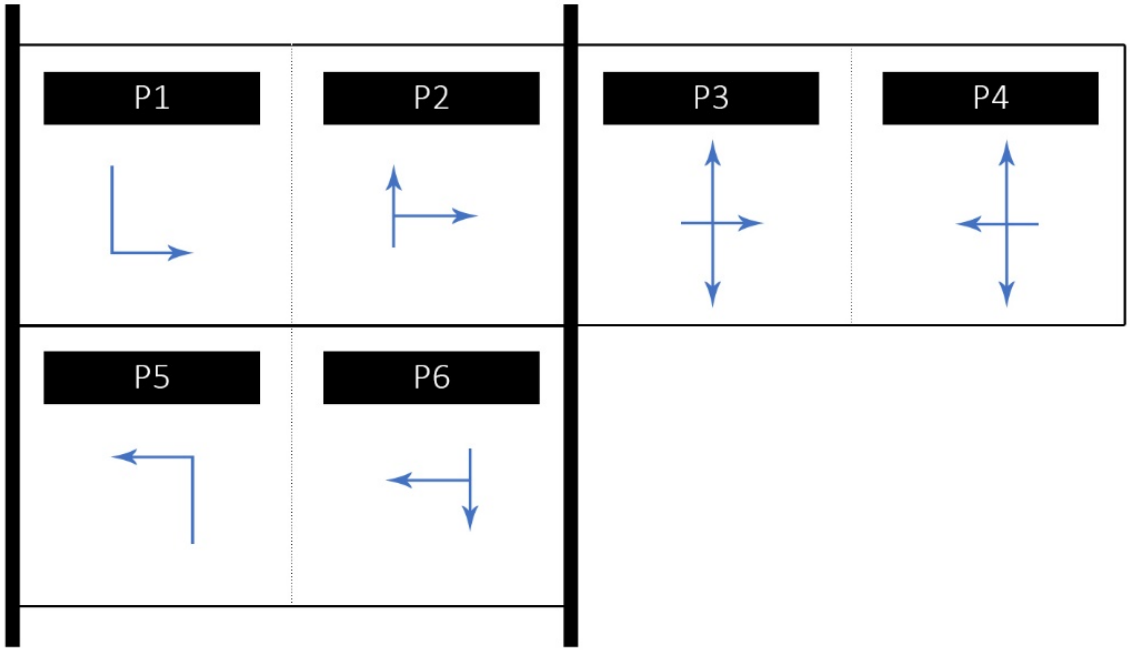

Figure 5: Ring Barrier diagram for National Electrical Manufacturing Association (NEMA) phasing

Intersection Geometry and Signal Phasing

As shown in Figures 1 and 2, the intersection has four approaches. The northbound and southbound approaches each have six lanes. The eastbound approach has two lanes with a right turn lane diverging from the right lane and the westbound approach has three lanes. Figure 2 shows the detailed lane configuration as well as the placement of 18 different detectors where there is a detector for each lane. Each detector provides data that includes traffic volumes, speeds, and occupancies as well as vehicle actuations measured by detectors for each separate lane.

The data collected all spanned from 6 am until 10 pm when the signal was operated in actuated coordinated phase. This means that while the overall cycle length was predefined, there was what is called “floating green time” which could be allocated between the different signal phases. The overall cycle length was changing based on a predefined time schedule according to the time of day and day of the week.

Figure 3 shows the NEMA phasing ring barrier diagram. Phases 1, 2, 3 and 4 are all on ring 1 whereas phases 5 and 6 are on ring 2. The barrier (indicated by thick lines) separates the movements in the north-south direction from the movements in the east-west direction. The signal timing plan starts with the left turns for the north-south directions followed by the through and right traffic. Overlap is allowed to occur between phases 1 and 6 and between phases 2 and 5. For

the east-west direction, split phasing is employed where the eastbound approach is allowed to discharge all through, left and right movements followed by the westbound approach discharging all of its movements. Three pedestrian phases occur simultaneously with vehicle phases 2, 4 and 6. The timing of these phases walk and do not walk as well as the pedestrian actuations/calls and the pedestrian signal status are also reported as part of the input data. Not all these phases have to be served for each cycle, but phases can be skipped if there are no actuations occurring for the phase, which adds to the complexity of the signal state prediction.

Data Preparation

Once all historical data was gathered into the database, this step prepared the data to be used in the machine learning step. The data was queried from the database and converted to input files for the LSTM networks. Converting JSON data files into LSTM inputs was a multistep process that involved several data manipulation steps.

Creating Data Frame from JSON Files

JSON includes data in a hierarchical nested structure of variables. For example, the timing data would be separate from the detector or pedestrian data. The JSON file was a hierarchy of data structures that included different data. For each second of data, there was a JSON file that included all the data for this second in a hierarchical structure. The first step involved extracting all the relevant data from the JSON files and converting the hierarchical tree-like structure to a flat structure such that each second can be an entry in a table-like structure or a data frame. The second step was filtering the data since the JSON file format was unified for different intersections and therefore had some redundant, duplicated, or irrelevant data for our intersection which was omitted. It should be noted that some variables were present on some days but marked as missing on other days. Therefore, to be able to unify the data frame structure, all data had to be examined first before deciding on what variables to include. If a certain day had no value for the variable, then it had to be marked as missing.

Preprocessing Numerical and Categorical Variables

At this point data was converted from a large number of JSON files with one file per second to a more concise set of table-like data frames with one file for each day of data. The third step was to recode categorical variables into dummy variables. This involved recoding variables like “signal state” or “exit mode” or “detector actuation” into n-1 dummy binary variables where n is the number of possible states of the categorical variable. The fourth step was normalization of numerical variables to be between zero and one. This was achieved by subtracting the minimum value of the variable and dividing by the difference between the maximum and minimum values.

The maximum and minimum values for each variable were obtained from sampling several days of data throughout the study period and obtaining the absolute maximum and minimum values from these days. These maximum and minimum values were not only used for the training data, but also the same values were used for the validation and testing data. The reason this is important is that in a field implementation, the maximum and minimum values for the prediction will not be known. Therefore, the maximum and minimum values for the training data will have to be used for normalizing the variables of the field data to obtain the prediction. The only numerical variable that was handled differently was the time of day which is a cyclical variable that was coded into two variables which are $\sin(2\pi t/n)$ and $\cos(2\pi t/n)$ where t is the time of day in seconds and n is the total number of seconds per day. This mapped the time of day as a point on a circle which is common practice for deep learning data preparation. After this process each second of data was comprised of 187 variables which feed into the prediction.

Re-indexing Time Steps and Output Variable Computation

At this stage, all variables are in the correct format; however, some of the seconds in each day are missing and some are duplicated due to connection. This is rectified by re-indexing all the tables by using a unique id which is chosen to be UNIX time. UNIX time is the number of seconds elapsed since Jan. 1, 1970. By using the UNIX time as an index, all missing intermediate data could be filled in as missing data. The missing data was given a unique value "-1" as none of the variables had values below zero. This step is essential as the input to the LSTM network has to be time series equally spaced across time. The time step between subsequent data points had to be set to be exactly one second even if some of the data points in the time series are missing.

The next step was computing the output variable which is the time until the signal state changes from green to red or vice versa. The data included the state of the traffic signal at each time step but not the time until it changes state, so this involved looking ahead at a time horizon of 200 seconds and defining the time until the signal state changes. This was done for each of the 6 phases for the signal. It is one of the most computationally demanding data preparation tasks as it involves iterating up to 200 seconds in the future for every second where the time remaining cannot be deduced from the previous time step time remaining. In case the time remaining cannot be deduced from the next 200 data points either due to missing data or due to phase skips prolonging the switching time beyond 200 seconds, the switching time is marked as missing and is not used for training the model. The output variable is then normalized by dividing by 200 seconds (the maximum value where the minimum value is zero).

Generation of Sequences and Batch Processing

Once all data was arranged with consecutive time steps in seconds, the next step was generating

input for the LSTM network. The LSTM network was trained in batches where each batch was a three-dimensional data structure containing 1000 sequences and 1000 predictions. Each sequence was a two-dimensional data structure or a matrix of values. The matrix consisted of vertically stacked data points describing all the input variables for the duration of the past two minutes until the second where prediction is taking place. Each time step in this sequence contains the 187 input variables obtained from the JSON files. For each of these sequence matrices, the prediction is a 6-element vector which contains the time left for each of the 6 signal phases to change. The training data included a total of 2,164,000 sequences and their corresponding predictions, making 2164 batches of training data. Handling these sequences was very memory intensive as the sequences were 387 gigabytes of data and therefore required a very memory efficient operation to be able to shuffle all sequences and load them to the model in varying order.

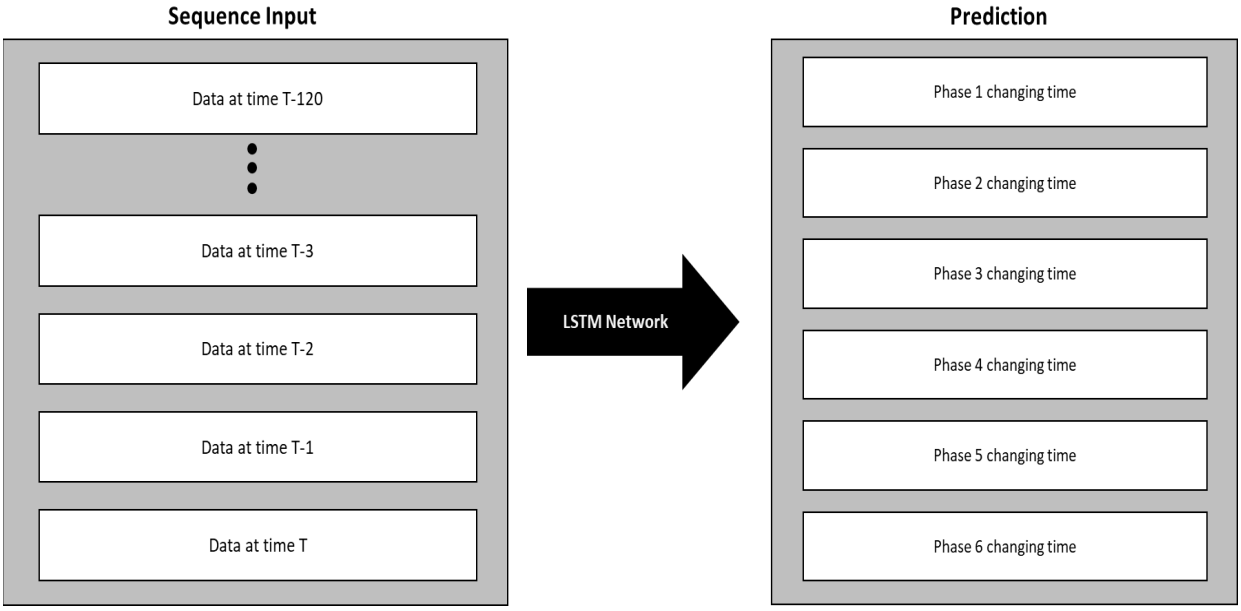

Figure 6: Sequences and Prediction Data Structures

Machine Learning

LSTM Networks

Recurrent neural networks (RNN) were introduced as a deep learning framework which includes temporal dependencies between temporal sequences of hidden layers. Recurrent neural networks, however, did not perform very well training on data with long-term temporal dependencies. It was critiqued for the issue of vanishing and exploding gradients while training (*34*).

LSTM model architecture was proposed as an alternative for typical RNN architecture because it is more capable of storage and access of long-term temporal dependencies. The

architecture, originally proposed in 1997, has been continuously improved and adapted into modern research paradigms (*35*). The architecture used in this paper is similar to that used by Graves in 2013 and depicted by the following equations(*36*).

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (2)$$

$$c_t = f_t c_{t-1} + i_t \, tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (3)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_{t-1} + b_o) \quad (4)$$

$$h_t = o_t \, tanh(c_t) \quad (5)$$

These five equations describe the LSTM cell basic architecture with input x_t for time step t. i, o, and f describe the input, output and forget gates, respectively. c refers to the inner cell which stores the information. h refers to the hidden state. The given architecture provides 11 sets of weights and 4 biases connecting between the different gates and the cell. Furthermore, the cell values and hidden states from previous time steps are all fed into the cell, input, output and forget gates allowing the neural network to adjust the weights and store only the meaningful temporal dependencies between the different time steps.

Model Formulation

The LSTM neural networks were built using TensorFlow Keras, which is a high-level machine learning package in Python. All the tested models consisted of five fully connected layers (Figure 7). The input layer had 187 nodes to process all data inputs. This was followed by an LSTM layer with N number of LSTM units where N is a value which was experimented with to find the best model. This was followed by a dense fully connected layer with rectified linear unit (ReLu) activation with N number of nodes. The model was set up as a regression output model where for each phase, the time remaining for the phase to change could be any value between zero and 200. This means that the output prediction was to be a linear combination of the outputs of the dense fully connected layer with ReLu activation and that the output layer was to have 6 nodes with linear activation. This allows the network to output the values of the time remaining for the state to change from red to green or vice versa for in each of the six signal phases. This means that the output is continuous variables which then have to be rescaled to 200 seconds and then approximated to the nearest second. The models all utilize an Adaptive Moment Estimation (Adam) optimizer, and different loss functions are experimented with including mean squared error, mean absolute error and mean absolute percentage error. The different models obtained are

compared against one another and prediction errors in the short term and long term of different models are used. More details of the different models used, and their performance, are discussed in the results and analysis section.

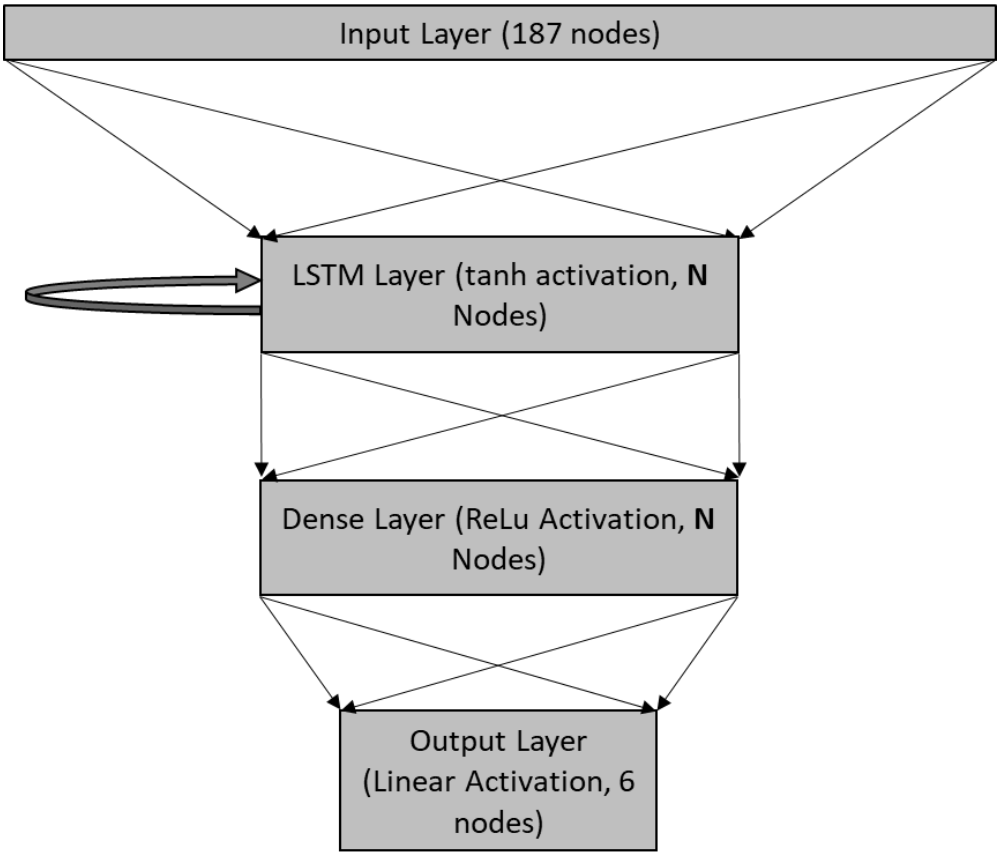

Figure 7: LSTM Network Architecture Used

Results and Analysis

Data Split

Cross validation was used for every model to prevent overfitting. The data was divided to three subsets. Training data is used to train the model. Validation data is used to check the performance of the model regularly during training to avoid overfitting and tune hyperparameters. Testing data is used to evaluate the performance of the finalized model. Similar to what would happen in a field implementation, the model was trained using 81 days of data between July 26, 2019, and the Dec. 10, 2019. The validation data used for tuning the hyperparameters of the models was used as a single day of data, which is Dec. 14, 2020. The testing data used for evaluating the performance

of the finalized models was Dec. 16, 2020. The validation and testing data were chosen to be for a date in the future outside the bounds of the training data. The reason for that is that the key interest lies in the ability of the models to generalize to future data rather than generalizing to historic data the model has not been exposed to. Furthermore, the use of a single day for validation and a single day for testing was done as a single day's worth of data contains a representative amount of data to assess the entire model performance. The fact that each second of data is considered a data point means that a single day of data contains enough data points to allow a model to be thoroughly assessed.

Exploratory Model Architecture Experiments

Two key model hyper-parameters were to be identified for the best model. The first is the value of N, which is the number of neurons, and the second is the learning rate to be used for the Adam optimizer. An exploratory model fitting experiment was done on 37 days from July 26 until Oct. 3, 2019. The data from Oct. 4, 2019, were used for validation. The 37 days included 1.1 million data points. Due to the large computational time associated with the sequence generation, loading data and model training, these models were limited to going over the data twice while monitoring the loss function and the performance on the validation data every time the model finishes processing 250,000 datapoints. A grid search was utilized to explore the hyper-parameter space for different combinations of learning rates and number of neurons per layer.

The choice of learning rates was on a logarithmic scale. For the choice of number of neurons, the value 187 is equal to the size of the input. It is then reduced twice by a factor of 4 to 47 and 12, respectively, and then the size of the output layer 6 is used. Mean relative error was used as a loss function and performance metric to compare different models in this case for two reasons. The first is that the absolute value of the error is less important the more the prediction horizon; so, for example, an error of 5 seconds in a 10 second horizon is much more than an error of 5 seconds in a 50 second horizon. The second is that previous papers which employed LSTM to travel time and traffic flow forecasting, which are the most relevant areas to our application domain, utilized it as a key performance metric (*29; 37; 38*). Its performance against other loss functions would be tested in the following section.

Table 2: Mean absolute percentage error performance on Validation Data

Learning Rate	Number of Neurons			
	6	**12**	**47**	**187**
1*10^-2	22.61	21.03	17.57	20.00
1*10^-3	28.56	33.00	20.18	27.36
1*10^-4	73.38	52.51	41.73	39.16
1*10^-5	223.27	269.53	196.74	108.81
1*10^-6	364.86	425.77	382.64	369.63

Table 2 shows the minimum reached mean absolute percentage error for different combinations of number of neurons per layer N and learning rate. One key insight that can be drawn from this analysis is that with the complexity and large computational time involved, choosing the correct learning rate is a key determinant of the quality of the model reached where the variation of the mean absolute percentage error over the varying learning rates is much larger than over varying number of neurons. Another observation is that the 47 neuron is better performing at both the 0.01 and 0.001 learning rates. Accordingly, the number of neurons would be set to 47 for further analysis.

Performance of Different Loss Functions

Model Description

The maximum prediction horizon for the models used is 200 seconds. This is 3 minutes and 20 seconds in the future and there are multiple uncertainties affecting predictions further ahead in the future. This property of having more certainty in predictions that are in the near future and less certainty in predictions in the distant future is common in many problem domains, and predicting traffic signal switching times is one of them. This necessitates the choice of a proper loss function

to reflect this property to be able to properly balance the prediction over short- and long-term prediction horizons. Three commonly used loss functions are examined which are mean squared error, mean absolute error and mean absolute percentage error, which is sometimes referred to as relative error. Another loss function is proposed by the authors, which is a modified version of the mean squared error which can be expressed as:

$$L = \left(y_{pred} - y_{true}\right)^2 * (1 - y_{true})^2 \qquad (6)$$

This function modified the mean squared error function by scaling it to be smaller as the true value of y is greater. It should be noted in this case that the true value of y has to be scaled between zero and one. This both uses the squared value of the error as well as scaling down the loss as the ground truth value of the time remaining is further ahead in the future similar to relative error. However, it provides more flexibility than relative error as the value of relative error increases indefinitely as the ground truth approaches zero whereas the proposed loss function would be approaching the sum of squared error as the ground truth value approaches zero.

All four loss functions are used to fit the entire dataset with the model discussed in Figure 5 with N=47. According to the findings from the exploratory networks, the learning rate was set as 0.01. Despite the Adam optimizer having its own adaptive learning rates based in momentum, an additional learning rate decay is added which multiplies the learning rate by 0.3 for every epoch where there is no improvement in the validation loss. This allows the learning rate to go down an order of magnitude for 2 epochs with no improvement. All 2,164,000 data points are fitted using each of the four loss functions discussed over 10 epochs allowing the model to assess the performance on validation data each epoch. For every loss function, the model with the best loss over validation data is selected and then the best model over the 10 epochs for each loss function is compared against those of other loss functions.

Model Performance Comparison

After fitting the models and deciding the best model for each loss function in terms of validation loss, the final models are evaluated by applying them to the testing data. The testing data contains 29578 sequences and predictions. In terms of overall model performance over the 200-second prediction horizon, the proposed model has the least absolute errors (Figure 8). Mean squared error and mean absolute error functions follow with mean square error having higher probability to be within 5 seconds from the ground truth but absolute error having higher probability to be within any error range bigger than 5 seconds. Mean absolute percentage error has the worst overall performance over the entire prediction horizon even though it has a higher probability than mean absolute error of being up to 2 seconds from ground truth.

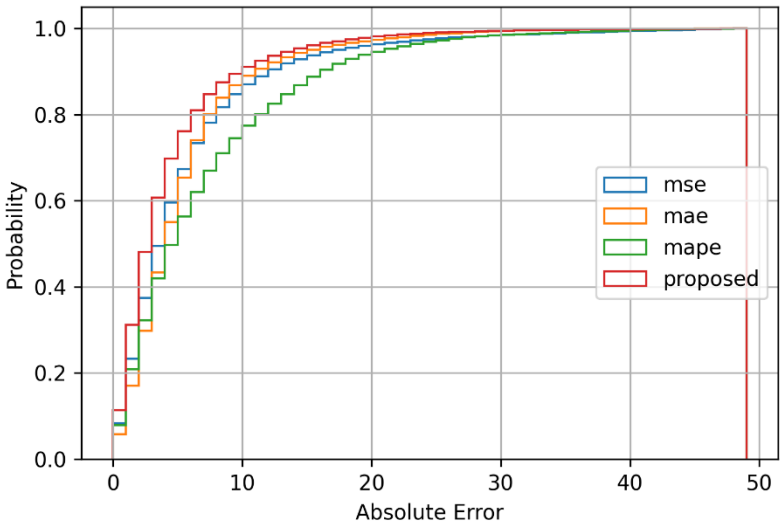

Figure 8: Comparison of the model error cumulative distribution for different loss function

If the performance at different prediction horizons is compared between the models, the role of the loss function in optimizing the predictions becomes more evident. Both mean squared and mean absolute errors do not value how far into the future the prediction is. This means that an error of 5 seconds results into the same mean squared error and mean absolute error whether the difference is between 10s prediction and 15s truth or whether it is between 100 seconds prediction and 105 seconds truth. This results in the performance of these two models being much better in the long-term predictions. Mean squared error has the lowest error for any prediction more than 100 seconds into the future (Figures 9-12) (Table 3). Mean absolute error has slightly lower error than the mean square error in the 80 second horizon and slightly higher error after 80 seconds.

The mean absolute percentage error (MAPE), on the other hand, heavily sacrifices long-term predictions for short-term prediction. It has the lowest error in the 20 second horizon and outperforms mean squared error and mean absolute error up to the 60 second horizon. It's also clear that it has much less distant outliers than other loss functions for values up to 40 seconds (Figure 10). This, however, comes at the expense of significant reduction in longer term prediction accuracy.

The proposed model is a middle ground between the two extremes which provides an overall better fit for the data. Despite being slightly outperformed by the mean absolute percentage error for the 20 second prediction horizon, it outperforms it on every other horizon. Furthermore, it provides the lowest error for up to a 100 seconds prediction horizon and is slightly outperformed by the mean squared error in the 100 second to 120 second horizon.

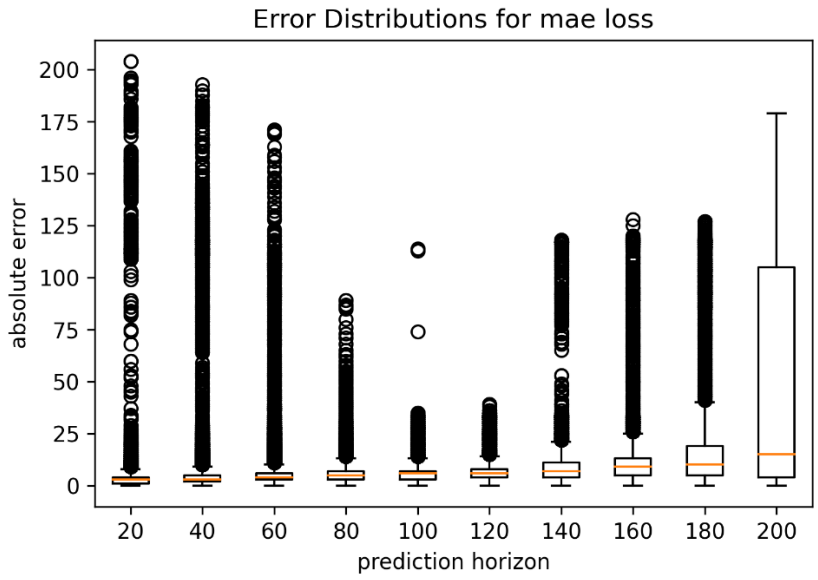

Figure 9: Boxplots for absolute error distribution over different prediction horizons for mean absolute error function best model

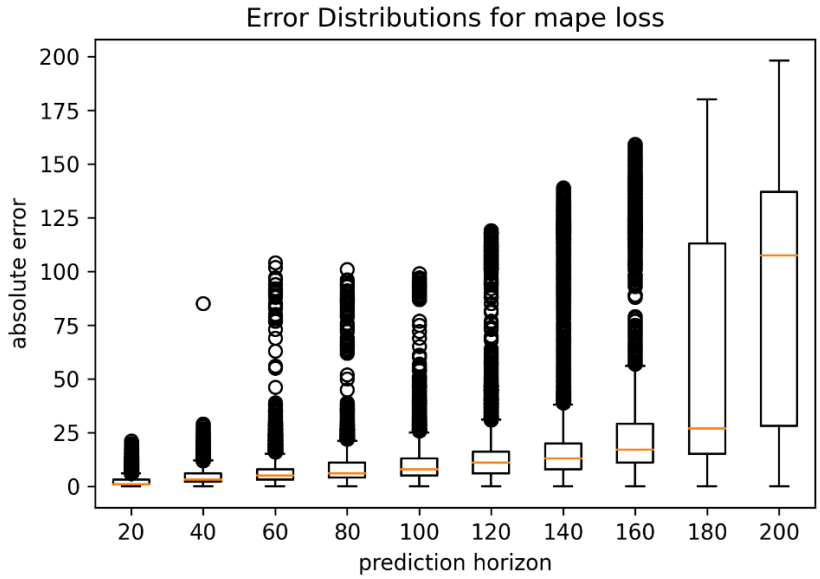

Figure 10: Boxplots for absolute error distribution over different prediction horizons for mean absolute percentage error function best model

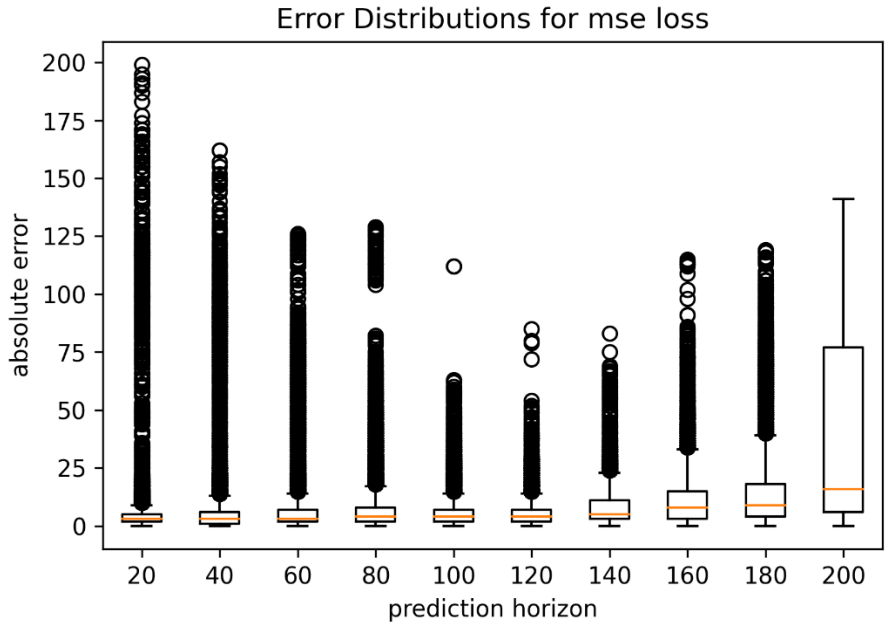

Figure 11: Boxplots for absolute error distribution over different prediction horizons for mean squared error function best model

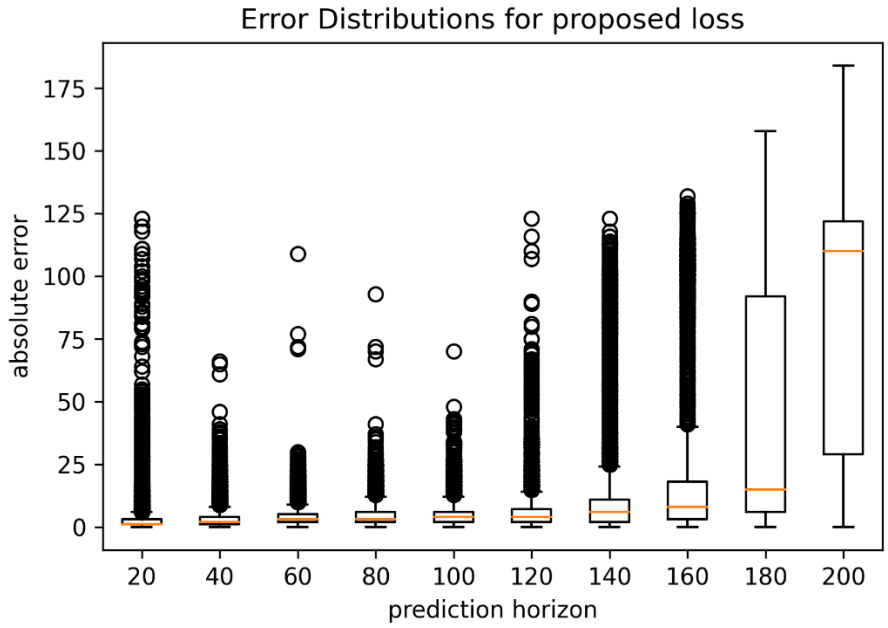

Figure 12: Boxplots for absolute error distribution over different prediction horizons for proposed error function best model

Table 3: Mean absolute error values over different prediction horizons for the best models using each loss function comparison

Loss Function	Prediction Horizon (s)									
	0-20	20-40	40-60	60-80	80-100	100-120	120-140	140-160	160-180	180-200
MSE	4.31	6.15	7.10	7.48	5.89	5.61	7.82	11.93	18.88	37.49
MAE	3.61	5.94	6.52	6.18	6.10	7.20	9.63	11.97	22.51	46.26
MAPE	1.96	4.20	6.08	8.09	10.03	13.46	26.17	35.49	52.40	89.45
Proposed	2.13	3.38	4.20	4.68	4.71	5.74	13.27	23.22	40.59	89.35

PART 2: EFFECT OF LOOK-BACK PERIOD

Methodology

The study relies on a three-step research methodology that includes:

- Data gathering and preparation.
- Training LSTM models on the same data using different look-back periods.
- Testing on out of sample data and comparison of model performance

Data Description

The data used in part 2 is similar in format to that used in part 1. It expands on the previous part by using a total of 150 days of data as opposed to the 83 days of data in part 1.

Data Split and Preparation

The training dataset is composed of 120 days of data between July 2019 and January 2020. This includes a total of 1.8 million sequences of data learned by the model. A single day's worth of sequences (15,000 sequences) on Feb. 17, 2020, is used for validation. The testing set is composed of 30 days out of the sample data for which the model has not seen during training within the same date range of the training set to be able to assess prediction performance in a way that mimics the model being deployed in the field.

To be able to compare the effect of the model look-back time window on the prediction performance, different look-back time windows are used for training and testing the same LSTM model. The time step used is one second, which is the same as the maximum level of detail offered by the data. For each second, the data provides a 187-element vector of the data elements described in Table 1. The sequence input to the LSTM neural network is provided as a matrix composed of the past 3, 10, 30 or 60 seconds of data. This means that for the 30-second look-back model, a single input data point is a matrix of size 187 x 30 and an output vector showing the time left for switching for each of the six phases of the traffic signal.

The input data are arranged into small sequences in time followed by the prediction. The prediction includes the time left for each of the six phases to turn green if it is currently red or time for it to turn red if it is currently green. Switch times can range from zero to 200 seconds, but it is normalized to be between zero and one. This is a very long prediction horizon, so in assessing model performance, the next 20 seconds is focused on rather than long-term predictions for the entire 200-second prediction horizon. The length of the sequences (time window to look in the past) is altered between models, so models were developed that used a time window of 3, 10, 30 and 120 seconds, respectively. A depiction of the sequence generation process is provided in figure 13.

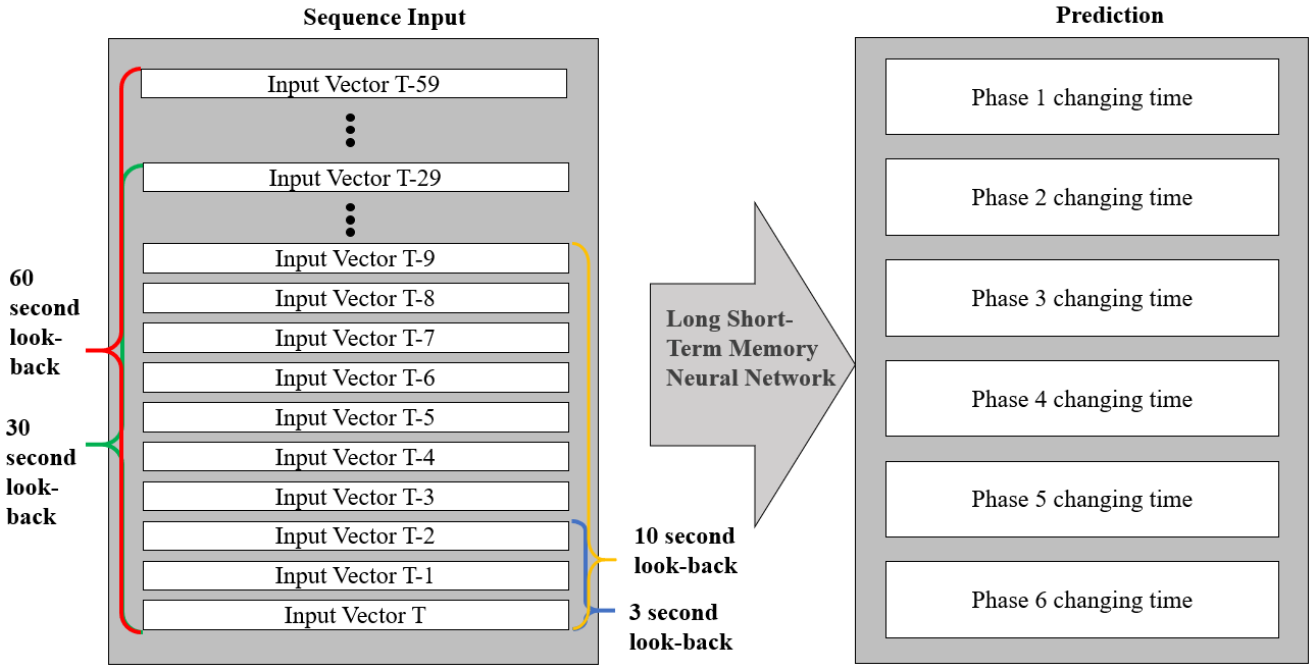

Figure 13: Sequences and input time steps

Model Formulation

The model architecture used is adapted from the optimal architecture used in part 1 based on the architecture proposed by Graves in 2013 (*25; 36*). The same architecture is trained using different look-back time windows to show the effect of the look-back time window on the prediction performance.

The model architecture used is shown in figure 14. The input layer is followed by the LSTM recurrent layer, where the length of the input sequence affects the training time of the trainable parameters in the model shown in equations (1-5). LSTM layers have a tanh activation as this is the only activation which has a GPU implementation in Keras that is much more efficient than the CPU implementation. This is followed by a fully connected layer to take in the output from LSTM and aggregate it. This layer uses a sigmoid activation function to map the outputs from the LSTM layer to a value between zero and one. Finally, a fully connected linear output layer is used to linearly combine the outputs from the sigmoid layer to come up with the six switching times of the traffic signal.

The loss function used is the mean absolute percentage error (MAPE) or the mean relative error (MRE), which was proven to achieve the best performance for a 20-second prediction horizon in part 1 (*25*). The purpose of using MAPE is prioritizing predictions in the short term and giving much less priority to predictions in the long term. Knowing the switching time with a good accuracy directly upstream of a signal allows eco-driving systems to better adapt the vehicle speed profile to navigate the intersection in a fuel-efficient manner. For this reason, more attention is given to the model performance in the short term (20 seconds and shorter) than in the long term.

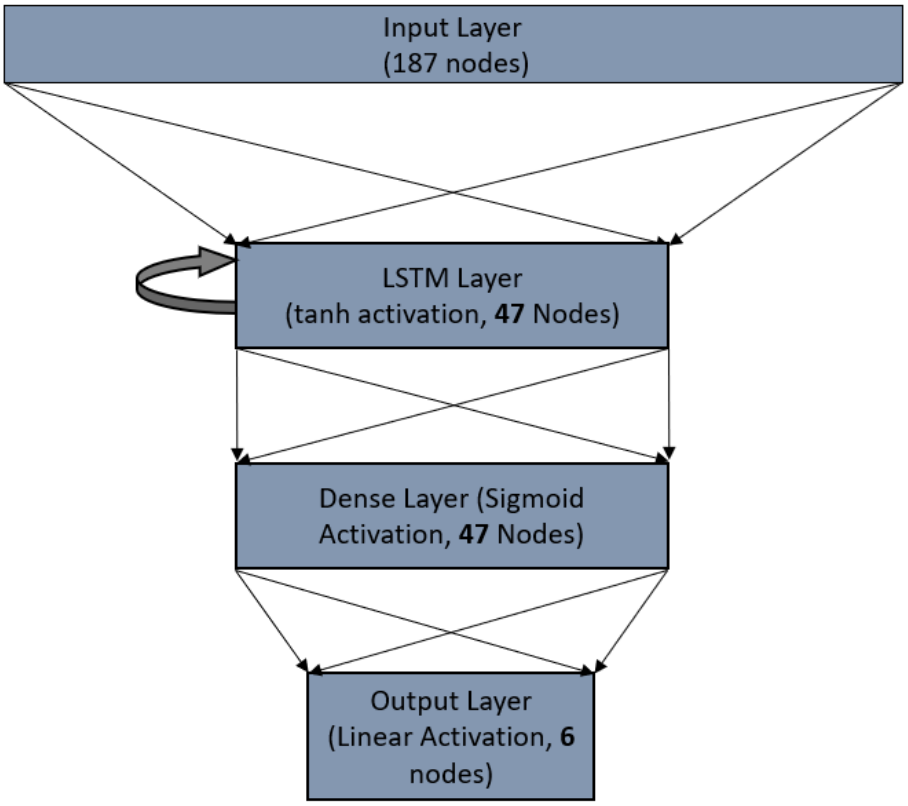

Figure 14: Model Architecture

Results and Analysis

Short-Term Prediction

Four models with the same architecture described in figure 14 are built and then run on the training dataset until they converge after 14 epochs. The models' performance on out of sample data is illustrated in the error distribution shown in figure 15.

In terms of the short-term prediction horizon of less than 20 seconds, all models are similar to one another in terms of performance despite having a very slight advantage for the model using the 10-second look-back window. Even the 3-second model performs fairly well. This indicates that the trends repeating themselves in the data even within a period of 3 seconds can be sufficient to obtain a good estimate of what is happening over the next 20 seconds.

The models perform well in predicting the switching time from red to green or vice versa. The models on average provide an exact prediction 62% of the time. On average, the model predictions are off by a maximum of only 1 second from the ground truth more than 80% of the time. The predictions are off by 2 seconds maximum from ground truth more than 89% of the time.

This shows a good reliability of the prediction which can be used for decision making for intelligent vehicles.

When accounting for the total computational time required for all models, using a smaller look-back time window can be a good option. This is because the computational time and resources including memory and processing power required for the models with a small look-back interval are significantly smaller and these models tend to converge significantly faster.

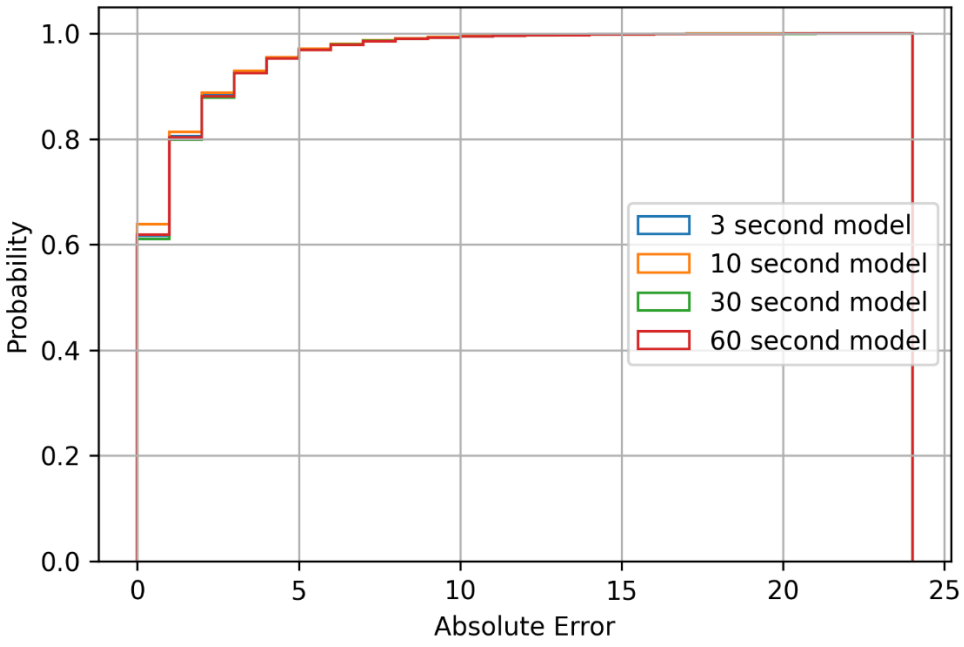

Figure 15: Short-term prediction absolute error distribution

Long-Term Prediction

The use of the mean relative error loss function makes this model much better suited for short-term prediction rather than long-term. In long-term prediction models, other functions such as mean squared error are much better suited and provide a better fit compared to the mean relative error.

In this case, however, it is beneficial to see how well the different models perform in terms of long-term prediction to explore any correlations between the look-back period and the model performance in long-term predictions. The error distribution for the models for the entire 200-second prediction horizon for the out of sample test is shown in figure 16. Unlike the short-term prediction where all models had a very similar performance, performance in long-term prediction

is better for the 60-second model followed by the 10-second model. The 30-second model produces a worse error compared to the 10-second model with a performance very close to that of the 3-second look-back period model.

The overall prediction performance in the long term is average, which is expected because of the use of the relative error loss function that focuses on the short-term and less on the long-term reduction in error. Overall, for the 60-second look-back model, prediction is on average within 9 seconds of the ground truth 80% of the time. For the 30-second, 10-second and 3-second lookback models, predictions are within 14 seconds, 11 seconds, and 14 seconds, respectively, from the ground truth 80% of the time. Given that the prediction horizon is 200 seconds, these values are reasonable though not particularly good.

It should also be noted that the models used provided a few unreasonable predictions (either less than zero or more than the 200-second prediction horizon). These distant outliers accounted for 0.8 percent of the total predictions and were filtered out as would be the case if they were in a field implementation of the model. One way to better deal with these outliers is to use model ensembles that obtain predictions from multiple models and take the average or use ML algorithms to weigh these predictions. This can help exclude one model when it comes up with an unreasonable prediction.

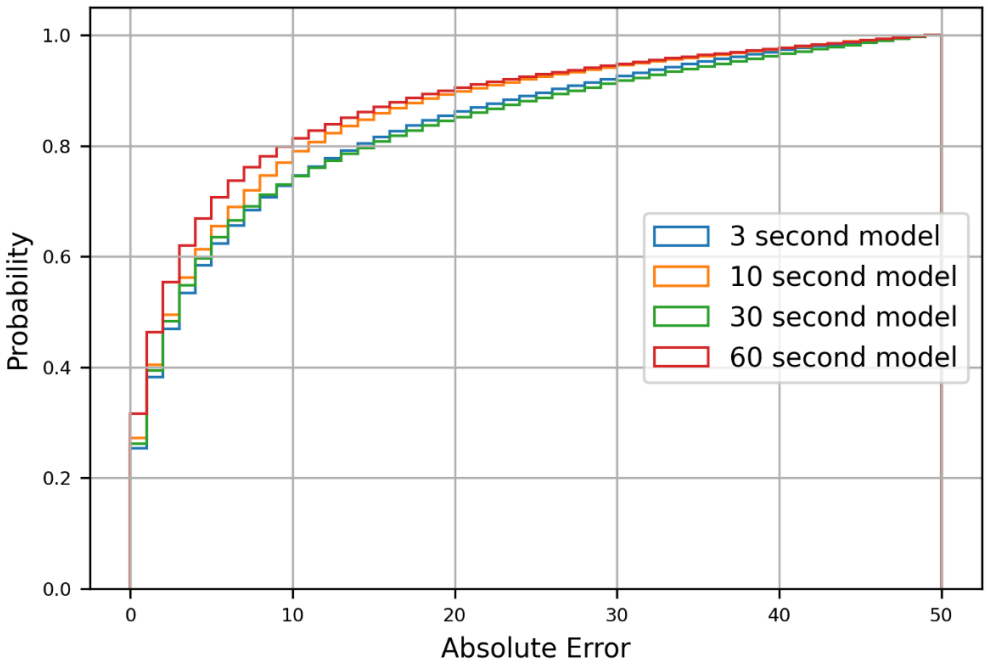

Figure 16: Long-term prediction absolute error distribution

Discussion

Comparing the long-term and short-term model performances considering different look-back periods, it can be discerned that short-term predictions are much less demanding for the LSTM network. Regardless of the look-back time window, all models result in good reliable predictions. Long-term predictions, on the other hand, are less obvious and do not demonstrate a specific trend. Specifically, while the error for a 60-second look-back period is much lower than any other period, the model performance considering a 10-second look-back period is better than that of the 30-second look-back period thus demonstrating that a longer look-back period is not necessarily better. The 3-second and 30-second look-back periods produce very similar errors; however, the 30-second look-back model is much more computationally expensive.

The overall recommendation for researchers using similar data is in case of conducting short-term predictions, start with shorter look-back periods and gradually increase the look-back and observe the effect on the overall model performance to achieve a model that has satisfactory performance and is computationally efficient. For long-term predictions, on the other hand, a more thorough grid search exploring different areas of the solution space can be examined.

The results show that there is no monotonic relationship between the look-back period and the overall prediction performance. This means that while increasing the look-back period can potentially add useful information to the model that it could use to detect a temporal trend, it can also add information that is not essential that can make it more difficult for the LSTM model to detect the more fundamental short-term trends in the time series data.

PART 3: EFFECTS OF REGULARIZATION AND GENERALIZATION PERFORMANCE

Methodology

This study methodology relies on the large reduction in travel demand created by COVID-19 to assess the short-term generalization of traffic signal switching time models (*39*). The study adopts a three-step research methodology:

- Data preparation and splitting.
- Development of different LSTM model variants.
- Assessing and comparing the different LSTM model variant performance on test data prior to COVID-19 and after COVID-19.

Data Preparation and Splitting

The data are split into three datasets. The training dataset covers 120 days of data that span the months from July 2019 to January 2020 similar to those used in part 2. For cross-validation during training, a single day including 15,000 data points is used and this corresponds to Feb. 17, also similar to part 2. The first testing dataset is meant to assess the performance of the model variants on test data before the occurrence of the COVID-19 pandemic and includes 30 days from the same period of the training data between July of 2019 and January of 2020. The second testing dataset, on the other hand, is new data and is meant to assess the temporal generalization of the model as well as generalization to lower traffic demand conditions due to the COVID-19 pandemic. It covers the 18 days between March 15 and April 1 shortly after COVID-19 was declared a national emergency on March 1. Figures 17 through 22 show the difference in traffic volume demand on a typical weekday (Tuesday) and a typical weekend day (Saturday) between the days before the pandemic and those right after COVID-19 was declared a national emergency.

Figures 17 through 22 show the drastic decrease in travel demand for all six phases of the traffic signals after COVID-19 was declared as a national emergency. Note that the travel demand is recorded only for times of the day when the traffic signal operates in actuated-coordinated mode, which is 8:30 am to 9 pm on weekends and 6:30 am to 10 pm on weekdays. These are the same times on which all the models are trained. The reduction in demand is more pronounced on weekends than on weekdays. Moreover, during weekdays, it seems that demand is significantly reduced for travel before 5 pm, when rush hour traditionally occurs. On the other hand, demand after 5 pm is not significantly reduced showing that travel behavior during weekdays after work remains mostly unaffected. The travel demand reduction is also highest for the through movements for Gallows Road, which corresponds to phases 2 and 6 where the travel demand is almost halved. Figures 1 through 6 show that after COVID-19, demand drops to unprecedented levels, which is an unconventional circumstance in which the ability of the LSTM model variants to generalize is tested.

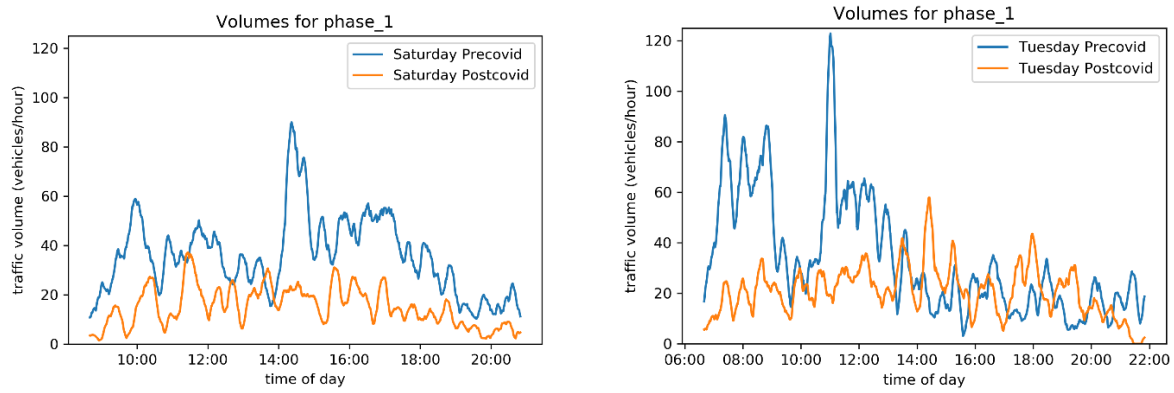

Figure 17: Weekday and weekend traffic volumes for phase 1 before and after COVID-19

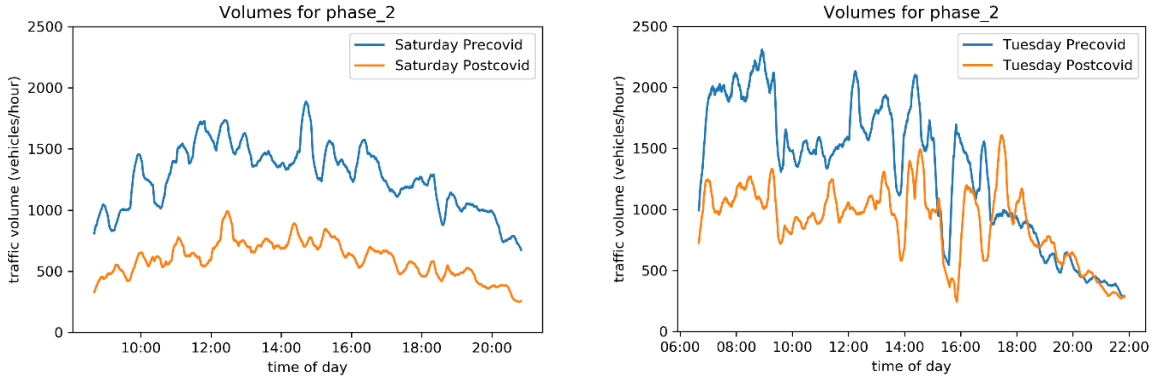

Figure 18: Weekday and weekend traffic volumes for phase 2 before and after COVID-19

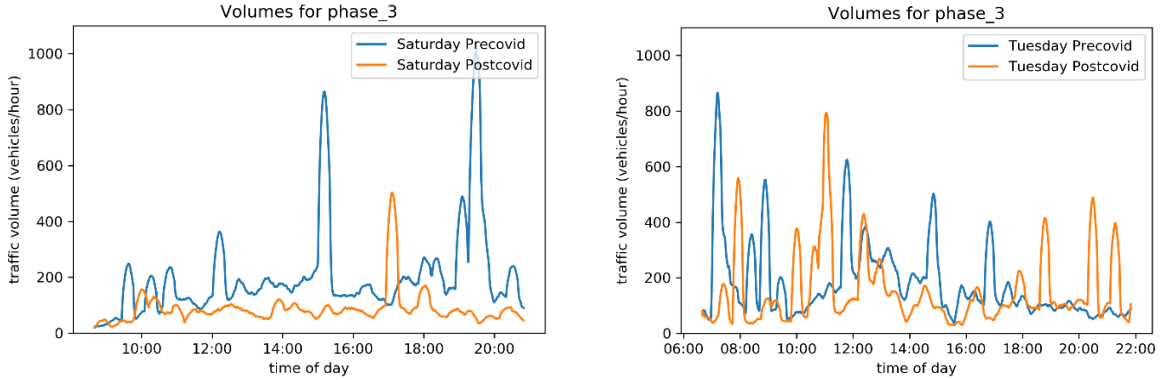

Figure 19: Weekday and weekend traffic volumes for phase 3 before and after COVID-19

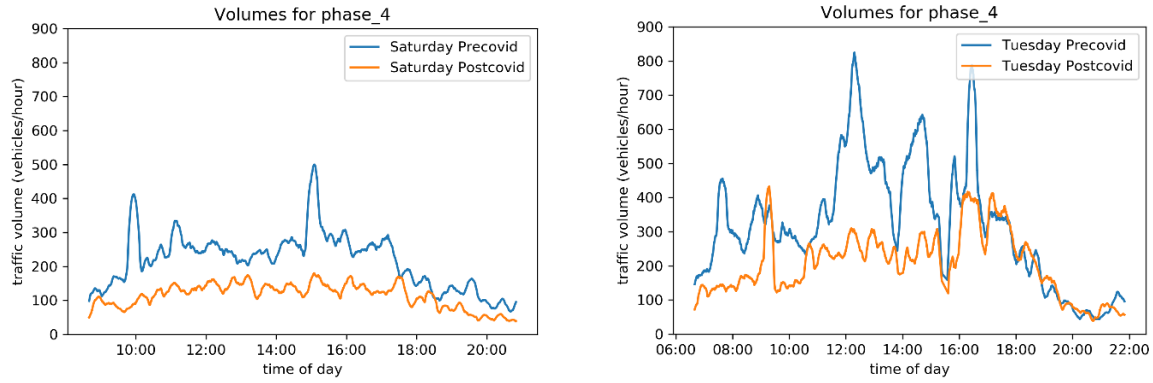

Figure 20: Weekday and weekend traffic volumes for phase 4 before and after COVID-19

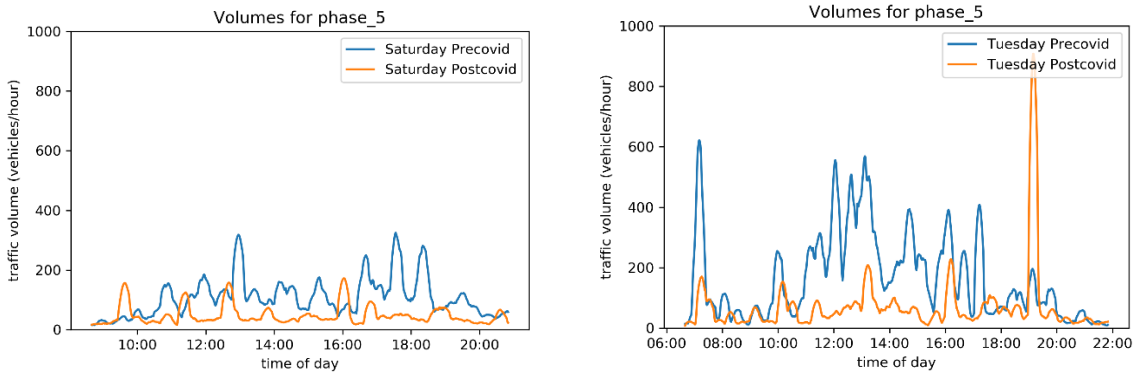

Figure 21: Weekday and weekend traffic volumes for phase 5 before and after COVID-19

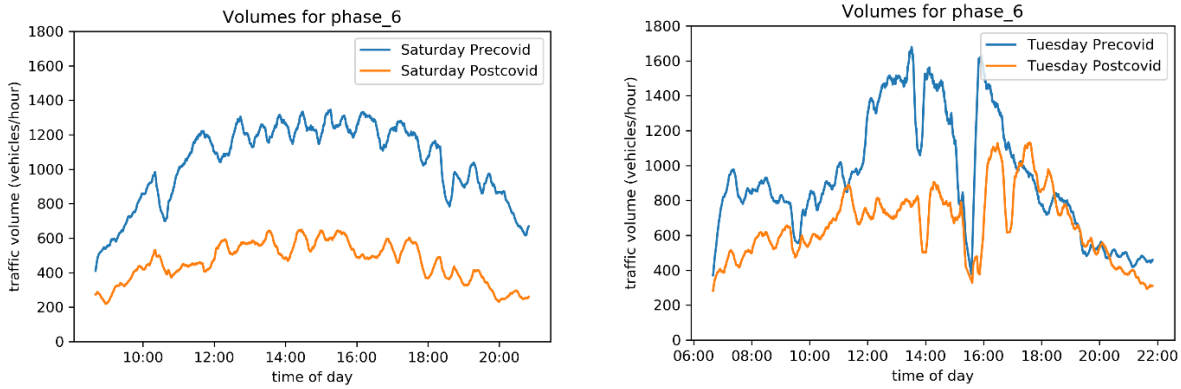

Figure 22: Weekday and weekend traffic volumes for phase 6 before and after COVID-19

LSTM Model Variant Development

The baseline model architecture used in this paper is similar to that optimized in part 1 (*25*). The two main differences in the model provided in this part are the prediction horizon and the use of

varying regularization techniques to create and assess model variants for generalization. The prediction horizon used for this study is 20 seconds, when the most gains in terms of fuel and energy efficiency from controlling the vehicle speed prior to reaching the intersection can be realized. It is also the prediction horizon where most previous studies achieved reasonable performance (*15; 24*). The loss function used is the mean absolute percentage error (MAPE) as it was found to provide the best results for short-term predictions (*25*). Similar to parts 1 and 2, the model was trained on a full 200-second prediction horizon; however, testing the model was limited to data points where the signal switching (from red to green or vice versa) was expected to happen within the next 20 seconds. This choice was made as training on long-term prediction was found to further improve the short-term prediction performance, so throwing away training data beyond the prediction horizon would be counterproductive for the model. The following section discusses the different regularization techniques used to create the model variants in further detail.

L1 and L2 Regularization Techniques

L1 and L2 penalty terms originated in statistical models where the L1 penalty is used for lasso regression and L2 penalty is used in ridge regression (*40*). The idea is that the L1 term is a term added to the loss function to penalize a percentage of the absolute value of the coefficients and L2 is added to penalize a percentage of the square of the loss function. In deep learning models, L1 and L2 can be used to penalize the weight and bias for each layer. Due to using the square of the weight, L2 penalizes weights less than 1 less than L1 and weights greater than 1 more than L1. Use of these regularization terms in the model loss function makes the model less likely to use very large weights and biases for a single input or activation. This makes models more stable and more likely to generalize better. The model variants using L1 and L2 use a ratio of 0.01 of the weights and biases absolute value for L1 and square value for L2.

Batch Normalization

Batch normalization is a technique in which the inputs to each layer (activations from previous layers) are normalized to reduce the changes in the distribution of the inputs to each layer during training (*41*). It is proven to accelerate the model training, reduce the sensitivity to model initialization state, and enhance generalization (*41; 42*).

Dropout

Random dropout of nodes within the deep learning model refers to omitting a random subset of nodes while training, as illustrated in figure 23. This subset that is omitted is continuously changed while training. This reduces the model chances of overfitting the training data or context-specific

adaptations within the network relying on a specific subset of features (*43*). In this study the dropout percentage is varied between 5%, 10% and 20%.

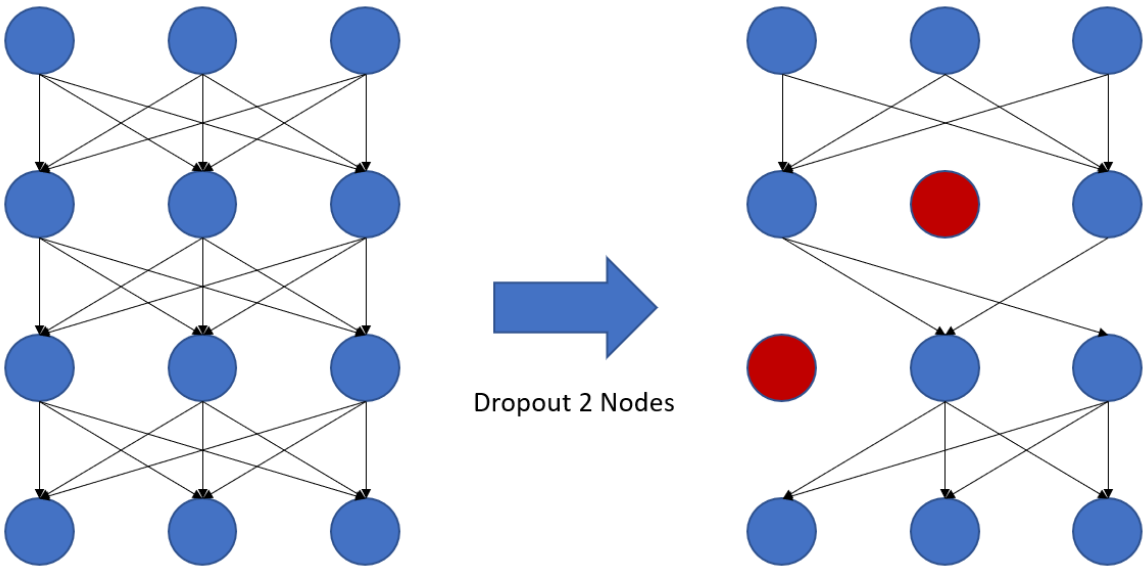

Figure 23: Dropout illustration

Model Variant Comparison

Based on the baseline model, nine other model variants are created using the discussed regularization techniques and compared to one another. The 10 model variants are created and assessed based on their prediction performance as well as generalization performance. The nine additional model variants use 5% dropout, 10% dropout, 20% dropout, batch normalization, L1, L2, combined L1 and L2, combined L1 and L2 with batch normalization and combined L1 and L2 with batch normalization and 10% dropout. Each model is trained on the training dataset and then used to predict using out of sample data from the duration of time before COVID-19 as well as after COVID-19 occurred.

Results and Analysis

Results

Each model is trained on the training dataset and then used to predict using out of sample data from the duration of time before COVID-19 as well as after COVID-19 occurred. Figures 24 through 27 provide a summary of the prediction performance of the nine different models with the baseline model performance provided in all figures as a reference. The graphs show a step function with the absolute error on the x axis and the cumulative probability of having this value of absolute error or less. The more the graph is shifted to the left-hand side the better the overall prediction

performance of the model. The graph is provided as a step function as both predictions and true values are discrete variables to the nearest second. Table 4 provides a comparison of the overall mean absolute error (MAE) of the different model variants prior to and after the lockdowns associated with the COVID-19 pandemic.

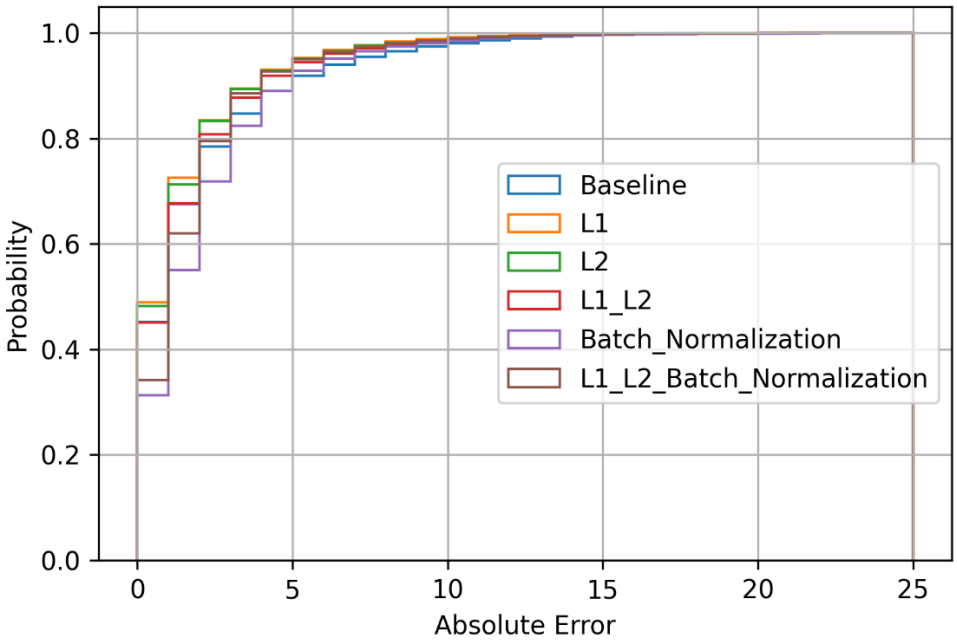

Figure 24: Absolute error distribution on test data before COVID-19 (Part 1)

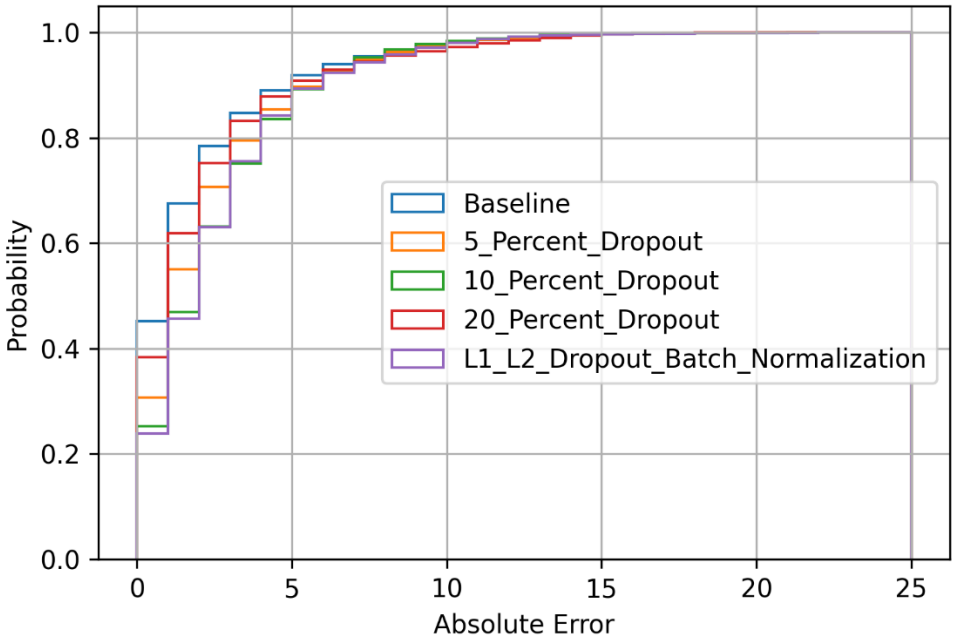

Figure 25: Absolute error distribution on test data before COVID-19 (Part 2)

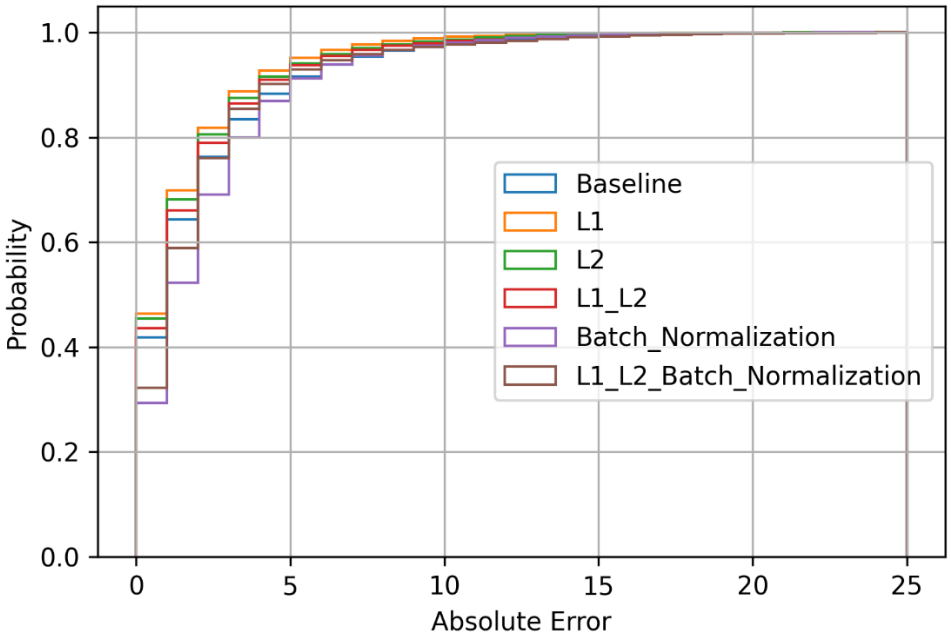

Figure 26: Absolute error distribution on test data after COVID-19 (Part 1)

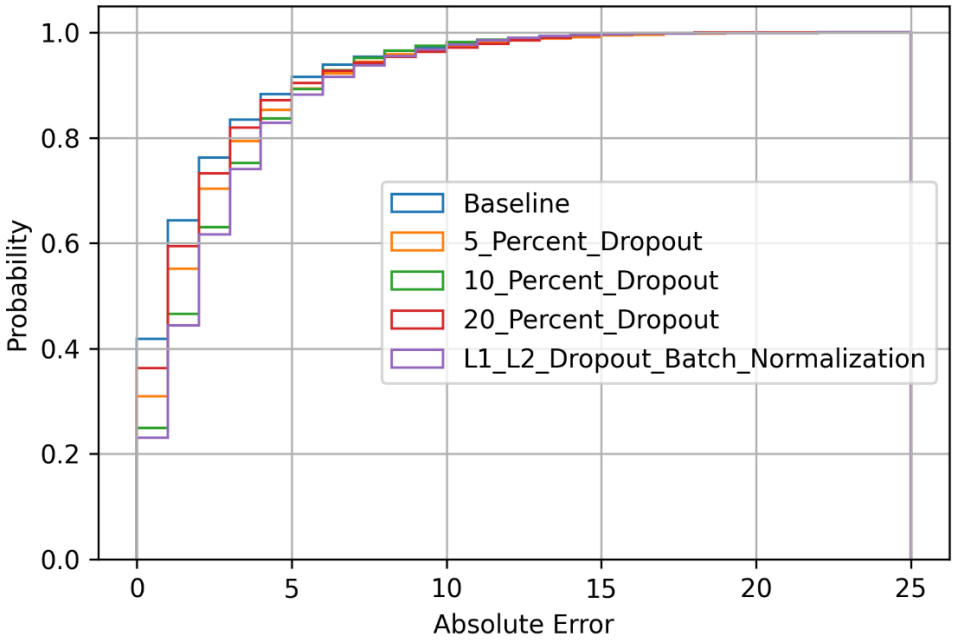

Figure 27: Absolute error distribution on test data after COVID-19 (Part 2)

Table 4: Model Mean Absolute Error prior to and after COVID-19

Model	MAE before COVID-19	MAE after COVID-19	Change (%)
5 Percent Dropout	3.61	2.88	-20.05
10 Percent Dropout	3.34	3.10	-7.08
20 Percent Dropout	3.56	2.60	-26.94
Batch Normalization	2.49	2.67	7.27
L1 L2	2.32	2.18	-6.02
L1 L2 Batch Normalization	2.36	2.46	4.06
L1 L2 Dropout Batch Normalization	2.97	3.08	3.77
L2	1.94	1.99	2.18
L1	2.06	2.05	-0.50
Baseline	2.78	2.34	-15.68

Analysis

Overall Model Performance

The model variants utilizing the L1 and L2 regularization parameters in the loss function provide the best overall model performance in both the test sets before COVID-19 and after COVID-19 with respect to MAE. The model using L2 regularization provides the lowest MAE of 1.94 seconds on the data prior to COVID-19 and 1.99 seconds on data after COVID-19. This is closely followed

by the model using L1 regularization with an MAE of 2.06 seconds on data before COVID-19 and 2.05 seconds on data after COVID-19. This is followed by the model variants using both L1 and L2 regularization, which have an MAE of 2.32 on data before COVID-19 and 1.18 on data after COVID-19. These are the only model variants that achieve a performance higher than the baseline performance even though the model employing batch normalization performance is very close to that of the baseline model.

Despite the L2 regularization model variant having a lower overall MAE compared to the L1 model, it should be noted that the L1 variant has a slightly more favorable error distribution where the L1 variant is within 2 seconds of the true value 83.5% of the time as opposed to 83.3% of the time on the test data before COVID-19. This is consistent with the performance on the data after COVID-19 as well where the L1 variant is within 2 seconds from the true value 81.8% of the time as opposed to 80.6 of the time for the L2 variant. This shows that the use of L1 and L2 regularization for stabilizing the weights and biases has a significant positive impact on the overall model performance. The L1 and L2 regularization significantly reduced the out of sample test error for both the test set before COVID-19 and that after COVID-19.

All model variants employing dropout layers have a significantly reduced performance compared to the baseline model. The dropout technique is meant to reduce dependency on a small proportion of the data or certain combinations of inputs to reduce overfitting. However, in the case of this training dataset, the overfitting is unlikely due to the huge size of the training dataset (1.8 million data points). This makes the model performance on out of sample data consistent with its performance on the training data and reduces the likelihood of overfitting even for the baseline model. This means that the use of dropout only makes the training process more complex without adding value in terms of generalization, leading to the reduction in performance seen in the results.

Complex model variants combining multiple regularization techniques seem to be outperformed by their simpler counterparts except for the case of the model variant combining L1 and L2 regularization with batch normalization, which outperforms the batch normalization only variant as L1 and L2 regularization improves the overall performance of the model.

Model Generalization Performance.

In terms of overall model generalization performance, the baseline model generalizes very well. The overall mean absolute error of the out of sample test data taken from after the COVID-19 time frame is 15.68% lower than the out of sample test data before COVID-19. The baseline simple model could achieve a prediction performance that does not reduce despite having a two-month lag between the training of the model and using it for prediction, which proves that the model is very robust to temporal variations. Moreover, the reduction in demand due to the COVID-19

pandemic did not affect the overall model performance which also proves that the model is robust to reductions in traffic demand. This result is counterintuitive as the lower the demand, the more difficult it becomes to predict vehicle arrivals and the more left turn phases and side road phases are likely to be skipped, which introduces error to the model.

Other model variants also perform very well in terms of generalization where the model performance before the COVID-19 pandemic seems to be consistent with that after the COVID-19 pandemic both in terms of MAE and error distributions. The worst performing model variant in terms of generalization was that employing batch normalization, and the MAE of this model increased by only 7.27% compared to its MAE before the pandemic. This model variant was within 2 seconds from the true value 71.9% of the time prior to COVID-19 and this was reduced to 69.1% after COVID-19. While this is one of the largest reductions in model performance among the tested model variants, the reduction in performance is still small. The second-worst model in terms of MAE is the one that uses all regularization methods combined, which increases its MAE by 3.77% after the COVID-19 pandemic. For this model, the model is within 2% of the true value only 63.1% of the time prior to the COVID-19 pandemic and 61.6% after the pandemic. This reduction is again very small. This shows that even the two worst generalizing model variants are still very consistent in terms of their performance after the traffic demand reductions during the COVID-19 pandemic. This result highlights the consistency and robustness of the model variants despite the two-month lag between training and testing as well as the reduced demand due to the COVID-19 pandemic.

CONCLUSIONS AND RECOMMENDATIONS

The use of LSTM neural networks allowed for a holistic data-driven modeling framework which takes into account controller logic, time of day, and vehicle and pedestrian actuation data. The presented framework is robust to missing data and provides useful insights into signal switching times whether it is time to green or time to red.

The report presented a step-by-step detailed methodology for data gathering, data preparation, training and tuning the LSTM models and testing and comparing different model architectures. The model was applied to the four-way traffic intersection between Gallows Road and Gatehouse Drive and provided predictions of up to 200 seconds in the future relying on the past two minutes of data. A comparison between different loss functions for training the LSTM network on our problem domain was conducted. In addition to the mean squared error, mean absolute error, and mean absolute percentage error, a new loss function was proposed. The overall performance of the proposed loss function was better than that of the conventional loss functions. The comparison between loss functions indicated the importance of the choice of loss function according to the desired results. For our data, if the priority is a very short-term prediction horizon

of less than 20 seconds, then the mean absolute percentage error is the best option. If the priority is prediction up to 100 seconds, then the proposed model is the best option. For long-term predictions of more than 200 seconds in the future, the mean square error is the best option. This study highlights the importance of choice of loss function to suit the usage of the prediction.

The report provides an evaluation of the effect of the look-back period on the overall model performance when using LSTM to compute traffic signal switch times. The results show that short-term accurate predictions less than 20 seconds can be obtained with a look-back time window as small as 3 seconds in duration. Long-term predictions, on the other hand, are more sensitive to changes in the look-back period and require a thorough search in the solution space to identify the best look-back period to enhance the model fit. The study also shows that the look-back period does not have to be larger than the prediction horizon or directly related to it. Instead, it depends more on the most important trends repeating themselves in the data with regard to the prediction. Given the results of part 2, the presented modeling framework is capable of being within 1 second from ground truth more than 80% of the time within a 20-second prediction horizon.

Finally, this project investigated the prediction and temporal generalization performance of LSTM neural networks with varying regularization parameters. Model variants for LSTM neural networks including dropout, batch normalization, L1 and L2 Regularization were used to train the model using 120 days of data, and their performance was evaluated on out of sample data prior to and after COVID-19 pandemic lockdowns. All model variants showed very consistent prediction behavior with very slight to no reductions in prediction performance between the testing on the data prior to the COVID-19 pandemic and the data after the COVID-19 pandemic lockdowns. This behavior suggests that the use of the 120 days of data was sufficient to avoid overfitting and produce a model with comprehensive understanding of the different traffic conditions and temporal changes.

The results of part 3 showed that L1 or L2 regularization parameters being added to the loss function significantly improved the model's overall prediction performance. Dropout layers significantly reduced prediction performance, and batch normalization did not affect the prediction performance. Generally, the simpler model variants outperformed the complex model variants combining several regularization parameters. This implies that researchers should be careful when using regularization parameters with traffic signal switching time predictions where adding more regularization parameters could be counterproductive. In the case of not having sufficient time for running an exhaustive search for the optimal model parameter setup, it would be best to train the model variants in a stepwise manner starting with the simplest model variants and adding to that.

While the proposed work is a good first step to testing the generality of the model, further

testing of the algorithm is needed on other intersections. Part 3 is intended to serve as a guideline for other researchers utilizing similar methods given the large cost associated with data gathering and high computational power needed to experiment with deep learning models. This effort can be indicative of the temporal generalization behavior expected when applying LSTM model variants at similar intersections to address questions about sufficiency of data and model performance under varying demand conditions.

This project provides a performance benchmark for future researchers using LSTM to predict traffic signal switch times. The results of this study are highly practical as the data collected are already being broadcast by the existing infrastructure without any extra investment in sensors or data collection devices. The modeling framework introduced in this paper can be deployed in the field and used to create models for any intersection. With the information provided by these models and the aid of stochastic control to account for prediction errors, automated and connected vehicles will be much more informed while navigating through traffic signals.

REFERENCES

1. Administration, U. E. I. *Use of Energy Explained: Energy use for transportation*. www.eia.gov/energyexplained/use-of-energy/transportation.php. Accessed July 13, 2021.

2. Barwise, Y., and P. Kumar. Designing vegetation barriers for urban air pollution abatement: a practical review for appropriate plant species selection. *Npj Climate and Atmospheric Science,* Vol. 3, No. 1, 2020, pp. 1-19.

3. Rakha, H., I. El-Shawarby, and J. R. Setti. Characterizing driver behavior on signalized intersection approaches at the onset of a yellow-phase trigger. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 8, No. 4, 2007, pp. 630-640.

4. Zhang, Y., C. Fu, and L. Hu. Yellow light dilemma zone researches: a review. *Journal of Traffic and Transportation Engineering (English edition),* Vol. 1, No. 5, 2014, pp. 338-352.

5. Wu, X., and H. X. Liu. A shockwave profile model for traffic flow on congested urban arterials. *Transportation Research Part B: Methodological,* Vol. 45, No. 10, 2011, pp. 1768-1786.

6. Rakha, H., and R. K. Kamalanathsharma. Eco-driving at signalized intersections using V2I communication.In *14th IEEE International Intelligent Transportation Systems Conference, ITSC 2011, October 5, 2011 - October 7, 2011*, Institute of Electrical and Electronics Engineers Inc., Washington, DC, United States, 2011. pp. 341-346.

7. Rakha, H., K. Ahn, and R. K. Kamalanathsharma. Eco-vehicle speed control at signalized intersections using I2V communication. *U.S. Department of Transportation, Tech. Rep.*, 2012.

8. Kamalanathsharma, R. K., and H. A. Rakha. Multi-stage dynamic programming algorithm for eco-speed control at traffic signalized intersections. Presented at Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference, 2013.

9. Chen, H., H. A. Rakha, A. Loulizi, I. El-Shawarby, and M. H. Almannaa. Development and Preliminary Field Testing of an In-Vehicle Eco-Speed Control System in the Vicinity of Signalized Intersections. *IFAC-PapersOnLine,* Vol. 49, No. 3, 2016, pp. 249-254.

10. Almannaa, M., H. Chen, H. A. Rakha, A. Loulizi , and I. El-Shawarby. Field implementation and testing of an automated eco-cooperative adaptive cruise control system in the vicinity of signalized intersections. *Transportation Research Part D: Transport and Environment,* Vol. 67, 2019, p. 18.

11. Stahlmann, R., M. Möller, A. Brauer, R. German, and D. Eckhoff. Exploring GLOSA systems in the field: Technical evaluation and results. *Computer Communications,* Vol. 120, 2018, pp. 112-124.

12. Karoui, M., A. Freitas, and G. Chalhoub. Impact of driver reaction and penetration rate on GLOSA. In *International Workshop on Communication Technologies for Vehicles*, Springer, 2019. pp. 16-26.

13. Yang, H., H. Rakha, and M. V. Ala. Eco-cooperative adaptive cruise control at signalized intersections considering queue effects. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 18, No. 6, 2016, pp. 1575-1585.

14. Suzuki, H., and Y. Marumo. Safety evaluation of green light optimal speed advisory (GLOSA) system in real-world signalized intersection. *Journal of Robotics and Mechatronics,* Vol. 32, No. 3, 2020, pp. 598-604.

15. Bodenheimer, R., A. Brauer, D. Eckhoff, and R. German. Enabling GLOSA for adaptive traffic lights.In *2014 IEEE Vehicular Networking Conference (VNC)*, IEEE, 2014. pp. 167-174.

16. Bingham, E. Reinforcement learning in neurofuzzy traffic signal control. *European Journal of Operational Research,* Vol. 131, No. 2, 2001, pp. 232-241.

17. Abdulhai, B., R. Pringle, and G. J. Karakoulas. Reinforcement learning for true adaptive traffic signal control. *Journal of Transportation Engineering,* Vol. 129, No. 3, 2003, pp. 278-285.

18. Abdoos, M., N. Mozayani, and A. L. Bazzan. Hierarchical control of traffic signals using Q-learning with tile coding. *Applied Intelligence,* Vol. 40, No. 2, 2014, pp. 201-213.

19. Lee, J., J. Chung, and K. Sohn. Reinforcement learning for joint control of traffic signals in a transportation network. *IEEE Transactions on Vehicular Technology,* Vol. 69, No. 2, 2019, pp. 1375-1387.

20. Abdoos, M., N. Mozayani, and A. L. Bazzan. Holonic multi-agent system for traffic signals control. *Engineering Applications of Artificial Intelligence,* Vol. 26, No. 5-6, 2013, pp. 1575-1587.

21. Xinhai, X., and X. Lunhui. Traffic signal control agent interaction model based on game theory and reinforcement learning. In *2009 International Forum on Computer Science-Technology and Applications, No. 1*, IEEE, 2009. pp. 164-168.

22. Abdelghaffar, H. M., and H. A. Rakha. Development and testing of a novel game theoretic de-centralized traffic signal controller. *IEEE Transactions on Intelligent Transportation Systems*, 2019.

23. Chu, T., J. Wang, L. Codecà, and Z. Li. Multi-agent deep reinforcement learning for large-scale traffic signal control. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 21, No. 3, 2019, pp. 1086-1095.

24. Weisheit, T., and R. Hoyer. Prediction of Switching Times of Traffic Actuated Signal Controls Using Support Vector Machines. In *Advanced Microsystems for Automotive Applications 2014*, Springer, 2014. pp. 121-129.

25. Eteifa, S., H. A. Rakha, and H. Eldardiry. Predicting Coordinated Actuated Traffic Signal Change Times using Long Short-Term Memory Neural Networks. *Transportation Research Record*, 2021, p. 03611981211000748.

26. Paul, B., S. Mitra, and B. Maitra. Calibration of Robertson's platoon dispersion model in non-lane based mixed traffic operation. *Transportation in Developing Economies,* Vol. 2, No. 2, 2016, p. 11.

27. Axer, S., and B. Friedrich. A methodology for signal timing estimation based on low frequency floating car data: Analysis of needed sample sizes and influencing factors. *Transportation research procedia,* Vol. 15, 2016, pp. 220-232.

28. Baluja, S., M. Covell, and R. Sukthankar. Approximating the effects of installed traffic lights: A behaviorist approach based on travel tracks. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, IEEE, 2015. pp. 2205-2212.

29. Duan, Y., Y. Lv, and F.-Y. Wang. Travel time prediction with LSTM neural network. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, IEEE, 2016. pp. 1053-1058.

30. Yuan, J., M. Abdel-Aty, Y. Gong, and Q. Cai. Real-time crash risk prediction using long short-term memory recurrent neural network. *Transportation Research Record,* Vol. 2673, No. 4, 2019, pp. 314-326.

31. Sameen, M. I., and B. Pradhan. Severity prediction of traffic accidents with recurrent neural networks. *Applied Sciences,* Vol. 7, No. 6, 2017, p. 476.

32. Fu, R., Z. Zhang, and L. Li. Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, IEEE, 2016. pp. 324-328.

33. Tian, Y., K. Zhang, J. Li, X. Lin, and B. Yang. LSTM-based traffic flow prediction with missing data. *Neurocomputing,* Vol. 318, 2018, pp. 297-305.

34. Pascanu, R., T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks.In *International Conference on Machine Learning*, 2013. pp. 1310-1318.

35. Hochreiter, S., and J. Schmidhuber. Long short-term memory. *Neural Computation,* Vol. 9, No. 8, 1997, pp. 1735-1780.

36. Graves, A. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

37. Zhao, Z., W. Chen, X. Wu, P. C. Chen, and J. Liu. LSTM network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems,* Vol. 11, No. 2, 2017, pp. 68-75.

38. Mackenzie, J., J. F. Roddick, and R. Zito. An evaluation of HTM and LSTM for short-term arterial traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems,* Vol. 20, No. 5, 2018, pp. 1847-1857.

39. Du, J., H. A. Rakha, F. Filali, and H. Eldardiry. COVID-19 pandemic impacts on traffic system

delay, fuel consumption and emissions. *International Journal of Transportation Science and Technology*, 2020.

40. Pereira, J. M., M. Basto, and A. F. da Silva. The logistic lasso and ridge regression in predicting corporate failure. *Procedia Economics and Finance,* Vol. 39, 2016, pp. 634-641.

41. Ioffe, S., and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.In *International Conference on Machine Learning*, PMLR, 2015. pp. 448-456.

42. Kukačka, J., V. Golkov, and D. Cremers. Regularization for deep learning: A taxonomy. *arXiv preprint arXiv:1710.10686*, 2017.

43. Hinton, G. E., N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.