

Traffic Forecasting Using LSTM Recurrent Neural Network

A Case Study of SR-405 Milepost 9

Jason Chen

Computing & Software Systems

University of Washington

Bothell, WA, USA

chench26@uw.com

ABSTRACT

Traffic flow study is an essential application that helps city planners to develop and optimize a transportation network. These data can indicate congested areas of a transportation network and suggest plausible solutions. With the rise of intelligent transportation system (ITS) in modern traffic flow studies, traffic data are aggregated in real-time and it is more cost efficient than before [1]. Due to the nature of the real-time intelligent system, the data are utilized in applications such as travel flow prediction to help facilitate the traffic. Other systems such as Google Maps, Bing Maps and Waze have utilized personal smartphones to collect real-time data to attain even more accurate travel time estimates. However, it is very costly to maintain the infrastructure such as Google Maps and the collection of personal location data is a very sensitive topic in today's climate where privacy is a major concern. In this project, I'm exploring the viability of using public resources, using the historical traffic data from the Washington State Department of Transportation and the National Weather Service to conduct a traffic flow case study using recurrent neural network to forecast traffic flow.

CCS CONCEPTS

• Machine learning • Time series analysis • Recurrent Neural Network (RNN) • Long short-term memory (LSTM)

KEYWORDS

Intelligent transportation systems (ITS), Traffic flow, Travel time prediction, Washington State Department of Transportation (WSDOT), SR-405

1 Introduction

Traffic flow study is an essential part of city planning as it can direct the city planners' attention to issue in a transportation network. Planners can take the traffic flow data and identify the bottlenecks, and further develop solutions to solve those bottlenecks. With the help of modern intelligent transportation systems (ITS), data are collected in real-time via telemetric and can be utilized for real-time applications [1]. A practical application of ITS is travel time estimates. The Washington State

Department of Transportation provides APIs for travel time estimates [2] [3].








State Route/ Interstate	Route Description	Distance (miles)	Average Travel Time	Current Travel Time	Via HOV (min.)
	Alderwood to Southcenter	29.42	29	29	N/A
	Alderwood to Southcenter	27.97	28	28	N/A
	Arlington to Everett	13.32	13	13	N/A
	Auburn to Renton	9.76	10	10	10
	Bellevue to Bothell	9.41	9	9	9
	Bellevue to Everett	26.06	26	26	26
	Bellevue to Federal Way	24.56	25	25	25

Figure 1: WSDOT Travel Times Page

Other systems like smartphone apps such as Google Maps, Apple Maps, and Waze use traffic data aggregated from personal devices are also able to provide travel time estimate in real-time. However, such data collection is problematic in couple of ways such as, personal privacy and cost effectiveness. Zang et al. discussed the challenges to come up an accurate prediction utilizing cellular traffic data [4]. Waze came up with an interesting concept which uses crowdsourcing as a mean to collect data. Such a platform requires active users, which Waze claims to have a community of over 115 million user [5]. Although, crowdsourcing traffic data cuts costs [6], to maintain the infrastructure for its operation and maintenance is still costly.

While travel time estimation is a practical application of traffic studies, I would like to focus on traffic forecasting, i.e. forecasting the rate of traffic flow and traffic volume. Google Maps also provides traffic flow information in real-time and forecasting. However, the forecast is a rough estimate of typical traffic, i.e. qualitative. Therefore, I'm conducting this case study to see if it is viable to train a machine learning model to produce a reasonable quantitative prediction on local data. I would like to breakdown this problem in twofold, the first is the data source, and the second one is the method. The historical data is downloaded from WSDOT's website and we will train a model

that will be able to forecast long-term traffic (24 hours in the future). In my literature survey, short-term prediction is usually defined between 5-45 minutes. We will also try to assess if variables, such as weather, season, holiday, major sport events, will influence the traffic flow. Taking the trained model along with the forecasts, we can hopefully produce a traffic predict that fits the traffic flow trend within reason.

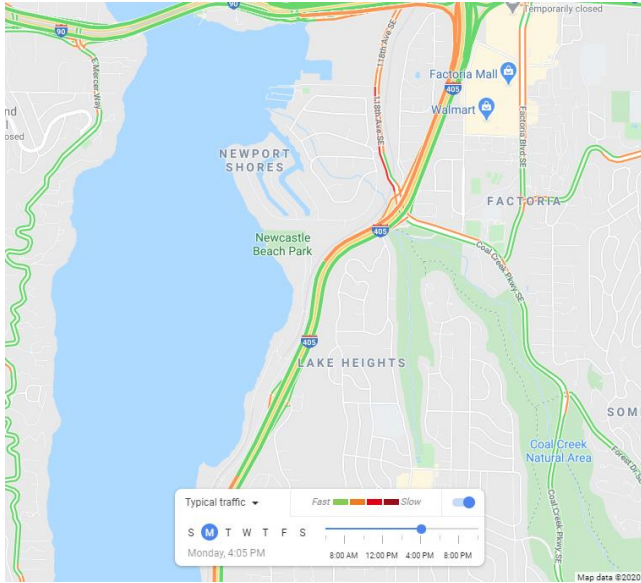


Figure 2: Google Maps traffic flow forecast [7]

2 Literature Review

According to my literature survey, before neural networks were applied to traffic flow analysis, the analysis were done using statistical regression. Common techniques include non-linear time series analysis ARIMA and Kalman filtering [8] [9]. Time series model are applied because traffic are periodical. In the other technique, Kalman filtering, a floating car is used for state estimation. Sherif Ishak and Haitham Al-Deek conducted their research on I-4 in Orlando, Florida and they have accounted for the differences between weekday traffic and weekend traffic. Their conclusion is that their short-term prediction model is more accurate when it is more congested [8]. Chien and Kuchipudi also utilized Kalman filtering and they conducted their research on the eastbound of NYST [9]. They have discovered that even at the same time on consecutive weekdays, the vehicle speed is inconsistent and the authors attribute the inconsistency to the daily average speed derived from historical data. They concluded that in order for the prediction to be accurate, the data must be taken from previous time interval (such as data collected 5 min earlier). Drawing from Chien and Kuchipudi's conclusion, I believe that recurrent neural network is a good candidate for this type of problem because RNN uses the previous information to predict the upcoming results.

Cheng et al. claimed that Kalman filter model has low accuracy and further proposed the gradient boosting decision tree (GBDT) model which utilizes decision trees and varying the weight of the trees based iterative training [10]. However, the dataset they worked on is simulated via VISSIM and they haven't accounted for variables such as weekdays/weekends differences which could be a major variable. The idea of using GBDT is interesting because it simulates the randomness of traffic patterns, and yet it still uses decision trees and iterative training to achieve a convergence which is more deterministic compared to using random forest to simulate peak hour traffic.

Zhan et al produced a consensus ensemble learning model [11]. They combined multiple models and then weighted them accordingly. The five models they used are ARMAX, partial least squares, support vector regression, kernel ridge regression and Gaussian process regression. While the concept of using an ensemble model is interesting, I think the solution is too complicate. Up to this point, all of the methods that I reviewed, most of them only analyzed the historical data and real-time data, and some accounted for the difference between weekdays and weekends, but none addressed the variables for weather, holidays, or major events besides [12]. I think it is worth investigating the amount of effect that each of these element might have on the traffic flow, such as [13].

One of the advantage of nonparametric model is that it is able to account for high dimensionality, such as the accounting for the day of year, day of week, seasonal attributes and etc. Tian and Pan proposed using the Long Short-Term Memory (LSTM) variant of Recurrent Neural Network (RNN) to make short-term traffic predictions. According to Tian and Pan, LSTM addresses a couple of problems that parametric model such as ARIMA have, mainly the inability to capture the nature of the highly non-linear and stochastic nature of traffic [14]. In other words, in nonparametric models, the data is treated statically. While ARIMA is model that uses the previous data to inform short-term prediction, it does not account for the fact that previous state of the data might influence the future state. LSTM is composed of an input gate, an output gate and forget gate. Not only is LSTM is able to remember the information from the past, it is also able to discard memorized information that isn't as important. Tian and Pan were able to achieve good results with LSTM on short-term predictions and I would like to apply LSTM on long-term predictions for this case study.

3.1 Data Aggregation

This study is conducted in the great Seattle area in the Pacific Northwest. The Pacific Northwest is known for the abundant precipitation and more recently the increase of snow during winter seasons. Weather is a major elements that cause significant increase in congestion. PNW is known for an active crowd and when the weather allows, many commuters choose biking as their main mode of transportation and thus reducing the vehicles on the

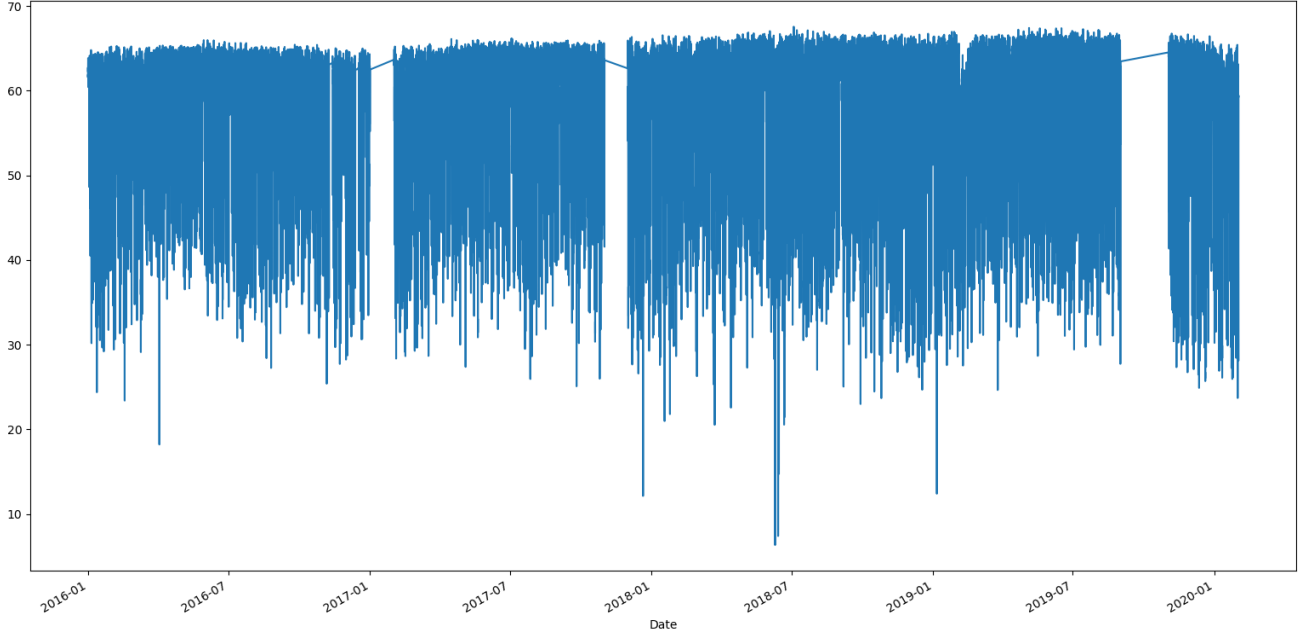


Figure 3: Missing traffic data causing wide gaps

road. Precipitation also decreases the traffic flow as the rate of accidents increases. Therefore, the season and the weather should impact the traffic flow. I will use weather data downloaded from reliable source. Seattle is also known for supporting its local sports team which often causes congestions on game days. Different days of the week might impact traffic, such as the Seahawk game day which happens on Sundays or Mariners game day on week nights and weekend afternoons. I will also consider the game days as inputs.

3.2 Traffic Data

There is no readily processed data for this project so I processed my own data. This project is based on the Interstate 405 (SR405/I-405) in Renton, WA. I downloaded the data via WSDOT's traffic portal [15]. The data is collected hourly at their permanent traffic recorder (PTR) site D1 which is located at milepost 9.26A. The data contains hourly speed per speed bin (5 mpg difference per bin, 19 in total, ranging from 0-5 mph to 90-95 mph), hourly volume, and volume by vehicle type in both directions. The traffic data spans from 2016-01-01 to 2020-01-31 and is about 4 years of traffic data. However, there are gaps in the data. I interpolated the smaller gaps (less than 4 rows), and discarded the data from the bigger gaps.

I used Python and Pandas dataframe to process the traffic data. The traffic speed file is the file that I used. I split the file into northbound and southbound files. I converted the raw traffic data into Pandas time series and added hour, day of week, and day of year as separate columns. They are added as separate columns so

they can be used as input later. LSTM will be able to recognize which day of the week it is. I then summed up the hourly traffic for each bin. Using the total traffic per hour, I calculated the hourly average speed using the Time Mean Speed formula:

$$\text{Time Mean Speed} = \frac{\sum_i^n \text{mean speed}_i}{\text{Total traffic}} \quad (1)$$

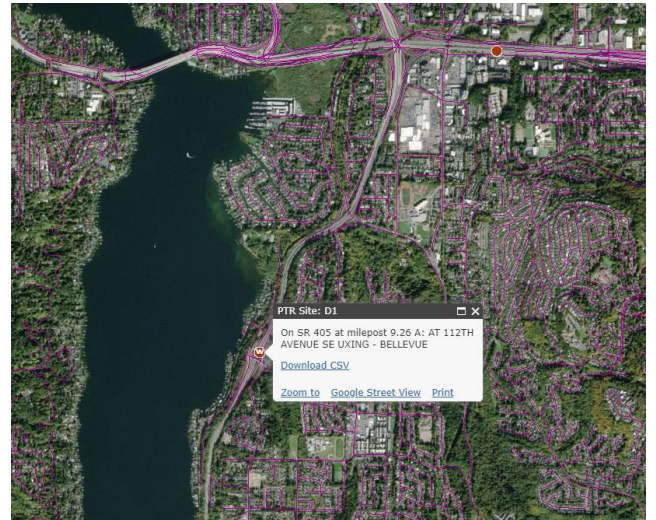


Figure 4: Permanent Traffic Record Site D1 [16]

Date time	Day of Week	...	0-5	...	35-40	40-45	45-50	50-55	55-60	60-65	...	Total	Avg Speed	Temp	...	Percip
2016-01-21 05:00:00	3	...	0	...	608	772	561	500	525	530	...	4062	50.47	45	...	0.02
2016-01-21 06:00:00	3	...	0	...	1113	548	425	534	321	108	...	4601	42.02	45	...	0.05
2016-01-21 07:00:00	3	...	0	...	696	557	465	328	259	25	...	4248	37.76	45	...	0.06
2016-01-21 08:00:00	3	...	0	...	565	371	470	232	268	17	...	3355	38.08	46	...	0.05
2016-01-21 09:00:00	3	...	0	...	488	374	321	140	204	0	...	3207	35.42	45	...	0.04
2016-01-21 10:00:00	3	...	0	...	776	797	650	410	406	290	...	3950	45.67	47	...	0.05
2016-01-21 11:00:00	3	...	0	...	389	785	946	652	377	369	...	4166	48.74	46	...	0.05
2016-01-21 12:00:00	3	...	0	...	97	300	880	1016	829	478	...	4293	54.91	48	...	0.08
2016-01-21 13:00:00	3	...	0	...	0	2	109	674	1376	1055	...	4191	60.60	47	...	0.02

Table 1: Final data sample - SR405 Southbound, 2016-01-21 (Thursday) 5am to 1pm. Table has been abridged to fit the page.

3.3 Weather Data

The weather data is collected at the weather station at Renton Municipal Airport which is about 8 miles southwest of PTR D1. I originally downloaded the data from the University of Washington weather portal [17]. However, there were 12 larger gaps in the data and they were beyond reasonable repairs. Each gap is around couple days and the gap happened about 12 times throughout 4 years. I was worried that the non-sequential data would affect the LSTM so I discarded the UW portal dataset. According to my research, sometimes the instruments aren't able to activate when there isn't enough stimuli. I tried the National Oceanic and Atmospheric Administration [18] and I found a different set of data. The NOAA dataset was very complete besides very few missing entries, about 1400 entries out of 35808 entries and the missing data weren't sequential. The weather data is also collected from the Renton Municipal Airport. I crossed checked the NOAA dataset with the UW portal dataset and they were similar though not complete identical. I assumed that the data from the NOAA is reliable.

The NOAA Renton Muni airport dataset contains 31 attributes and the full description can be found in 3505doc.txt. Examples of the attributes include precipitation in different intervals, temperature, visibility, etc. The interesting thing about the data is that is usually collected on 53rd minute of each hour and sometimes multiple times an hour. Due to the nature of the data, I used Pandas time series to resample the data and taking the max value of each hour. I choose max value instead of mean value

because the most important feature is precipitation and it is accumulative. The other attributes wouldn't be change as much within the same hour. I also shifted the data time ahead so instead of 53rd minute of each hour, the data is on the hour. I only picked the 9 most important attribute from the weather data and joined it with traffic data.

3.4 Final Dataset

I downloaded Seattle Mariners baseball schedule but it isn't in hourly format so I excluded it from the final dataset. Holidays are accounted for by the day of year feature and hopefully LSTM will be able to learn and detect the patterns of major sports events (Mariners/MLB, Seahawks/NFL, Sounders/MLS). I wish I had access to construction data and accident data but I could not find them on WSDOT's website. The final dataset contains a total of 33 features, including date time, day of year, day of week, hour of day, 19 speed bins, total traffic, average speed, and 9 weather data. Data is spanned from 2016-01-01 to 2020-01-31 and split into northbound and southbound files.

4 Method

Upon inspecting data (table 1), you can see the stochastic and non-linear nature of the data. The data is a sample of the final dataset, taken on 2016-01-21 from the morning to the afternoon. Only considering total traffic and average speed, while the total traffic remained around 4000 per hour from 5am till 7am but the average speed has decreased. Later in at 9am, there are even less

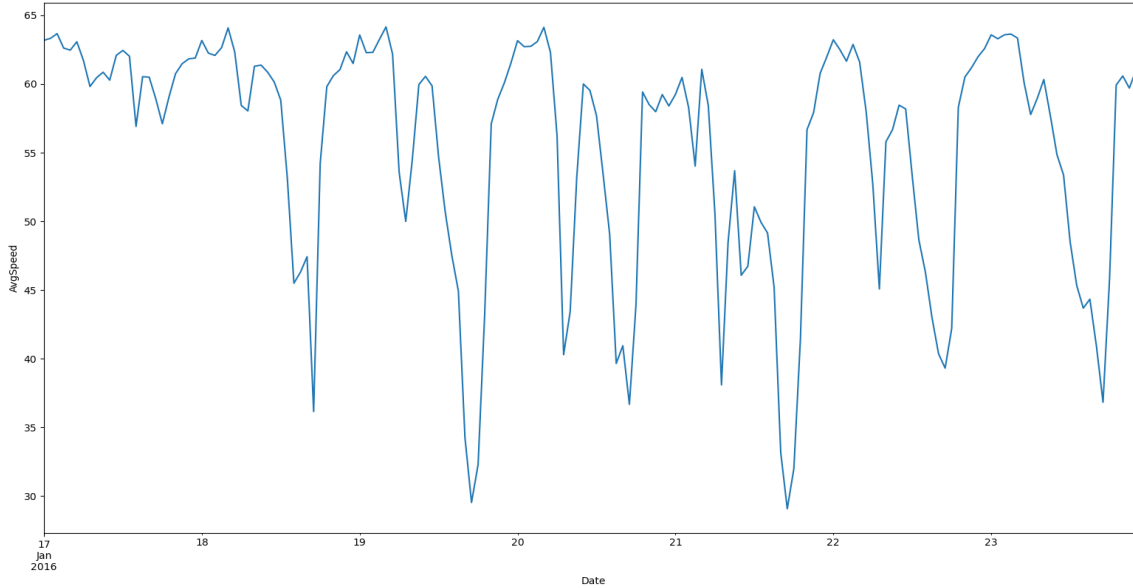


Figure 5: Average southbound traffic per hour from 2016-01-17 to 2016-01-23

traffic than at 7am, and yet the average speed is slower. This is because the highway is congested at this moment and causing less vehicle to pass through the PTR site. We also observe the 2 hour lag it takes for the average speed to drop from 50 mph to 35 mph though the total traffic remained roughly above 4000 vehicles per hour from 5am to 7am.

Inspecting the data visually in figure 5, you can identify the periodicity of the data. There are some irregular noise in the signal but generally the afternoons have the lowest average speed throughout the day. 2016-01-19 (Tuesday) and 2016-01-21 (Thursday) have the lowest average speed out of the week while the rest of the weekdays have similar daily low around 35 mph. Recurrent neural network (RNN) is good candidate for traffic prediction between it uses the past data as input. As we can see, the traffic congestion is caused by buildup of traffic and therefore, there is a lag until the average speed is impacted. The RNN variant that I'm using is LSTM. LSTM has a memory cell unit that is able to prevent the exponential decay of the loss function and thereby maintaining long term memory. The memory cell consists its own input gate, output gate, and modulation gate for the input. The memory cell also contains a forget gate to discard information that becomes less relevant. The memory cell feature could be helpful in detecting the traffic slow down lag.

I will use support vector machine (SVM) as a baseline for comparison. Tian and Pan have also used SVM compared to their LSTM model [14] but their model was trained for short-term predictions (5-15 min). I will train my model for long-term prediction, i.e. forecasting of the 24 hour. I will compare SVM's ability to forecast long-term traffic against LSTM. The library that

I'm using for SVM is the sklearn support vector regression model (SVR). For both SVR and LSTM, I will train the models without the traffic data, i.e. no total traffic, no speed bins, and only using the date/time and average speed. The speed bins and total traffic are heavily correlated to the average speed. The average speed and total traffic are derived from the speed bins. The more traffic there is in the lower speed bins, the lower the average speed will be. When we forecast traffic, we don't have the future total traffic and speed bins data. We will only know the date and hour of day, as well as the recent past traffic data. I will also train both the SVR and LSTM models using all features from final dataset to compare the accuracy. I will also train the both models with and without weather data to see impact weather has on traffic prediction.

5 Experiments Details

The shape of the final dataset is 35808 entries by 33 features. The 35808 entries contains 4 full years and 1 months of hourly data ((4 years \times 365 + 1 leap year day + 31 days from January 2020) \times 24 hours). The processed weather data doesn't contain any gaps but the traffic data does. By transitive property, the final dataset has the same gaps as the traffic data. I dropped all entries that are missing any data and there're 33840 entries left. The data is split into 30000 entries for training and 3384 entries for validation, roughly a 10-to-1 ratio. The input data for both models are scaled to between 0 and 1. For the error function, I'm using mean absolute error function for both SVR and LSTM. The validation data used to score the models are the same. I'm plotting the data from the last two weeks of January 2020.

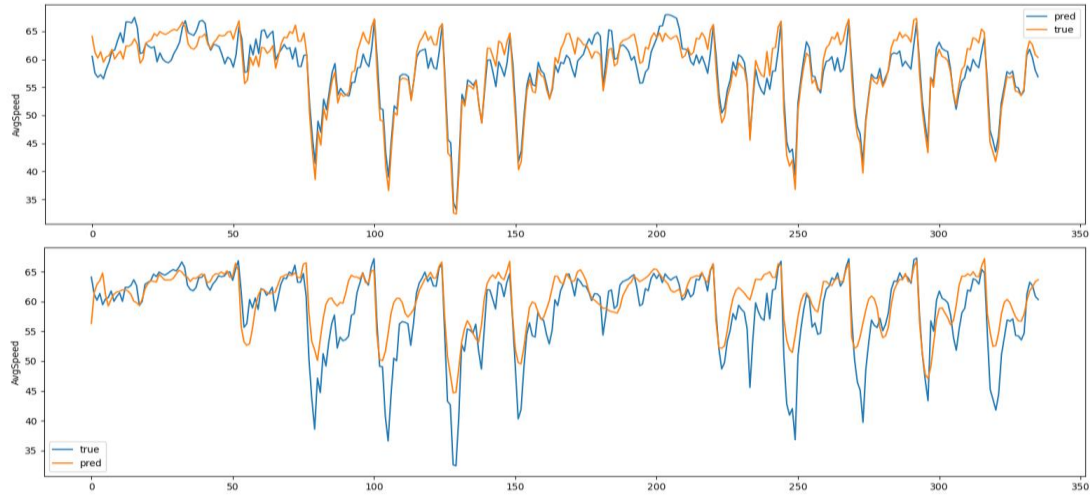


Figure 6: Top – SVR with all features; bottom – LSTM with all features

5.2 SVR

The SVR model uses a radial kernel and it only requires less than 1 second to train. SVR is cross validated using 10 fold cv SVR achieved 0.051 validation accuracy with all features but dropped to 0.064 when excluding the speed bins. Removing the weather data and excluding the speed bins provided similar results of 0.063.

5.3 LSTM

I'm using the TensorFlow Keras library's LSTM model. My model contains a single layered LSTM as input with the default tanh activation function and a single-layered output. I'm using a week's worth of data for LSTM to lookback. I tried a month of lookback but I ran out of memory so I wasn't able to see if there it increases the performance. Otherwise, a week of lookback trains reasonable fast. Due to the dropped data, the data isn't fully sequential and it's concern that it will impact the LSTM network. I first trained the data using the longest sequential data which is

the sequence from 2016-03-15 to 2016-11-08. It contains 5735 entries of data. I compared the validation loss with the full range of data and observed that using full range dataset provides slight performance increase. LSTM with full dataset and all feature is able to achieve 0.040 validation loss. A minuscule performance increase over the SVR model.

5.4 Experiments Results

If we forecast the traffic accounting only the date time and weather data features as input, because we don't have future traffic data at the time of forecasting, LSTM slightly out performed SVR. Scaling the MAE validation loss back to the original scale, LSTM only improved 1 mph over SVR, and SVR took faster to train. I was hoping that the LSTM would provide more improvement over SVR. From figure 6, we can see that the LSTM model follows the peak of the ground truth better while SVR is more accurate at predicting valleys. We are more concerned with how slowed when forecasting traffic, therefore, SVR is the winner in this case.

Model	Range of data	Model validation loss	Test validation loss	Including speed bins?	Using weather data?	Training time
SVR	all	0.064	0.064	no	yes	0.077 s
SVR	all	0.051	0.041	yes	yes	0.116 s
SVR	all	0.052	0.063	no	no	0.063 s
LSTM	all	0.035	0.046	no	yes	72 s
LSTM	all	0.027	0.040	yes	yes	114 s
LSTM	all	0.058	0.058	no	no	69 s
LSTM	2016-03-15 to 2016-11-08	0.031	0.039	yes	yes	95 s

Table 2: Training results and time

6 Conclusion

I spent a lot of time on finding reliable data and processing the data. At the end, I am still skeptical regarding the validity of the weather data from NOAA because I was anticipate a significant impact on the results. However, with or without the traffic data, it doesn't seem to make a big difference on the prediction. Judging from the outcome, it's either a placebo that Seattle traffic is severely impacted on rainy days or I haven't fully analyzed the correlation. There is also the possibility that the people base their commute decision on weather forecast instead of real weather conditions. I wasn't able to find historical weather forecast data. Although, I did download Mariners game day data, I didn't incorporate it into this study because it isn't hourly data and the location of the case study isn't on the main route of Mariners fans commute to and from the stadium. The parts of the road that is most impacted by game day is downtown Seattle and the I-90 bridge. I really wish LSTM provided more significant improvement over SVR. I learned that SVM/SVR is still a very powerful classification tool even comparing to the more complicated neural networks. I think the LSTM can still be tweak with more layers and hyper-parameter tuning. I barely scratched the surface of LSTM. It seems like the valleys of the LSTM predictions in figure 6 are quite reserved and the performance might be able to improve if the model can approximate those valleys better. I would look more into LSTM and its application in traffic forecasting. Other technique that I thought would be worth investigate is combination of Fast Fourier Transformation and RNN used by UBER [19].

REFERENCES

- [1] Agachai Sumalee and Hung Wai Ho, 2018. Smarter and more connected: Future intelligent transportation system. In *IATSS Research*, 42(2), pp. 67-71. Retrieved from <https://doi.org/10.1016/j.iatssr.2018.05.005>.
- [2] WSDOT. 2020. Intelligent Transportation Systems (ITS) Operations | WSDOT. Retrieved from <https://www.wsdot.wa.gov/travel/operations-services/its>.
- [3] WSDOT. 2020. WSSOT - Travel Times for Seattle all directions. Retrieved from <https://www.wsdot.com/traffic/traveltimes/default.aspx>.
- [4] Yunjuan Zang, Feixian Ni, Ziyong Feng, Shuguang Cui and Zhi Ding, 2015. Wavelet Transform Processing for Cellular Traffic Prediction in Machine Learning Networks. In *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, pp. 458-462. <https://doi.org/10.1109/ChinaSIP.2015.7230444>.
- [5] Waze. 2020. About Waze: Connecting Drivers with Live Traffic Maps. Retrieved from <https://www.waze.com/about>.
- [6] Mostafa Amin-Naseri, Pranamesh Chakraborty, Anuj Sharma, Stephen B. Gilbert and Mingyi Hong, 2018. Evaluating the Reliability, Coverage, and Added Value of Crowdsourced Traffic Incident Reports from Waze. In *Transportation Research Record* 2018, Vol. 2672(43), pp. 34-43. Retrieved from <https://doi.org/10.1177/0361198118790619>.
- [7] Google Maps. 2020. Retrieved from <https://www.google.com/maps/dir/@47.5691465,-122.2006403,14.25z/data=!4m2!4m1!3e0!5m1!1e1>
- [8] Sherif Ishak and Haitham Al-Deek, 2002. Performance Evaluation of Short-Term Time-Series Traffic Prediction Model. In *Journal of Transportation Engineering*, vol. 128(6), pp. 490-498. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2002\)128:6\(490\)](https://doi.org/10.1061/(ASCE)0733-947X(2002)128:6(490)).
- [9] Steven I-Jy Chien and Chandra Mouly Kuchipudi, 2003. Dynamic Travel Time prediction with Real-Time and Historic Data. In *Journal of Transportation Engineering*, vol. 129(6), pp. 608-616. [https://doi.org/10.1061/\(ASCE\)0733-947X\(2003\)129:6\(608\)](https://doi.org/10.1061/(ASCE)0733-947X(2003)129:6(608)).
- [10] Juan Chenb, Gen Li and Xianhua Chen, 2019. Research on Travel Time Prediction Model of Freeway Based on Gradient Boosting Decision Tree. In *IEEE Access*, vol. 7, pp. 7466-7480. <https://doi.org/10.1109/ACCESS.2018.8656924>.
- [11] Hongyuan Zhan, Gabriel Gomes, Xiaoye S. Li, Kamesh Madduri, Alex Sim and Kesheng Wu, 2018. Consensus Ensemble System for Traffic Flow Prediction. In *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, pp. 3903-3914.
- [12] Nicholas G. Polson and Vadim O. Sokolov, 2017. Deep learning for short-term traffic flow prediction. In *Transportation Research Part C Emerging Technologies* vol. 79, pp. 1-17. <https://doi.org/10.1016/j.trc.2017.02.024>.
- [13] Arief Koesdwiady, Ridha Soua and Fakhreddine Karray, 2016. Improving Traffic Flow Prediction With Weather Information in Connected Cars: A Deep Learning Approach. In *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508-9517.
- [14] Yongxue Tian and Li Pan. 2015. Predicting Short-Term Traffic Flow by Long Short-Term Memory Recurrent Neural Network. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pp. 153-158. Retrieved from <https://doi.org/10.1109/SmartCity.2015.63>.
- [15] WSDOT. 2020. PTR Sites Data Download. Retrieved from <https://www.wsdot.wa.gov/Traffic/API/PermanentTrafficRecorder/?siteId=D1&startYear=2019&startMonth=3&endYear=2019&endMonth=4&tables=TrafficSpeedByHour>.
- [16] WSDOT. 2020. Traffic GeoPortal. Retrieved from <https://www.wsdot.wa.gov/data/tools/geoportal/?config=traffic>.
- [17] UW Weather portal. 2020. Live From Earth and Mars – GRAYSKIES – Northwest Weather Resource. Retrieved from http://www-k12.atmos.washington.edu/k12/grayskies/nw_weather.html.
- [18] NOAA. 2020. Quick Links | National Centers for Environmental information (NCEI). Retrieved from <https://www.ncdc.noaa.gov/data-access/quick-links#dsi-3505>.
- [19] YouTube. 2020. Two Effective Algorithms for Time Series Forecasting – YouTube. From <https://www.youtube.com/watch?v=VYpAodcdFfA>.