Long term predictions for traffic forecasting

How does the accuracy degrade with time?

Bas Verlooy

Supervisor: Elena Congeduti

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Bas Verlooy
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, George Losifidis

Abstract

Traffic prediction plays a big role in efficient transport planning capabilities and can reduce traffic congestion. In this study the application of Long Short-Term Memory (LSTM) models for predicting traffic volumes across varying prediction horizons is investigated. The data used is collected by the municipality of The Hague for a single month. The study focuses on comparing the performance of the LSTM across different time horizons up to 10 hours in the future. To evaluate the performance of the LSTM models, two common evaluation measures are employed: Root Mean Square Error (RMSE) and Symmetric Mean Absolute Percentage Error (SMAPE). The baseline for the predictions is set at a 15-minute future forecast. Comparing the 1-hour prediction against the 10-hour predictions relative to the baseline RMSE, the RMSE increased threefold. However, the SMAPE first increases, but surprisingly after 6 hours starts to decrease again.

1 Introduction

Traffic forecasting is a critical component in the management and operation of modern transportation systems. Reliable predictions can reduce congestion but also improve traffic safety and efficiency, contributing to the overall performance of the city's traffic flow. However accurately predicting traffic flows over a long period poses significant challenges. The dynamic, non-linear nature of traffic systems influenced by a number of external factors such as the weather, time of day, and accidents make long-term prediction a difficult task . Accurate predictions over a longer time period can improve the transport planning capabilities [1] and reduce traffic congestion.

Long Short-Term Memory (LSTM) is a variant of a Neural Network (NN) that has often been used to forecast traffic [2]. Soon et al. (2022, page 20) stated about LSTMs in the context of traffic forecasting: "LSTM ... outperforms ARIMA, SVM, Stacked Auto-Encoder, Radial Basis Function Neural Networks, as well as RNN". Traffic forecasting is a time series prediction problem which means that previous data points are used to forecast a future value. LSTMs are able to capture temporal correlations, which makes them useful for traffic forecasting [2].

LSTMs are a type of recurrent neural network (RNN) that can learn and remember over long sequences, making them well-suited to handle temporal dependencies inherent in traffic data. This paper focuses on the use of LSTM networks in long-term traffic forecasting, providing an overview of their performance against different time horizons.

Other papers typically forecast traffic conditions for a 5-minute ahead timeframe of up to 60 minutes [2; 3], this study aims to give an overview of the prediction horizon from 15 minutes up to 10 hours. The research question of this paper is as follows: 'How does the LSTM model handle long horizon predictions and how does accuracy degrade with time?' The following two subquestions will help answer this:

1. How do the different evaluation metrics evolve when increasing the prediction horizon?
2. Which hours contribute the most to the change of each evaluation metric?

The data utilized in this study was gathered by the municipality of The Hague in November 2019. The data was collected by a total of 130 detection loops, also called sensors, embedded in the road surface. Although data for cars, bicycles, and trams were available, this study only focuses on car data. This choice was motivated by its relevance for comparison with existing literature and the significance of accurate predictions applied to car traffic. The data was aggregated by 15 minutes, also referred to as a timestep in this paper.

Previous research has been done, Licheng et al. (2019)[4] focused on predicting traffic 1 day in advance using a Deep Nural Network (DNN) and using external factors. Factors such as the weather, holidays, time of day, and day of the week are taken into account. The focus of this research was to experiment with different time intervals and see how the MAPE performed differently from an interval of 5 minutes up to 60 minutes. The results showed that an increasing interval would often result in a lower MAPE. Furthermore, they concluded that between 23:00 and 06:00 the MAPE was higher due to big relative differences in data when the traffic is low.

In [5] traffic similarities and repeatability were investigated with short and long-term predictions. The Mean Repeatability Degree (MRD) and Mean Similarity Degree (MSD) were used to predict traffic. They achieved a low Mean Absolute Relative Error (MARE), but the MRD and MSD are not going to be used in this paper.

This paper aims to contribute to 2 gaps in previous research, firstly to give an overview of how traffic prediction performs over different time periods. Secondly, to show the LSTM performance for roads with not as much traffic as on highways or interstates.

For many papers, the focus has primarily been on predicting traffic patterns within a short-term period. The definition of long-term prediction differs from paper to paper, some mention more than 30 minutes [6; 7; 8], while others refer to 1 day [4; 5]. A shorter time frame allows for responses to traffic congestion or incidents, which sometimes cannot be predicted over a long-time horizon, which is what makes long-term forecasting difficult.

Furthermore, the majority of existing research has centered around predicting traffic conditions specifically on highways or roads with high traffic volumes [2; 3]. These areas have been of particular interest due to the frequency of congestion and the necessity for effective traffic management and planning [9]. The dataset used is from an urban road and relatively contains a lot of periods with zero cars measured at 9.5% of the complete dataset.

The paper is structured as follows, section 2 details why LSTMs are useful for traffic prediction, introduces the problem formally, gives a description of the dataset used, and finally shows how the model is evaluated. In section 3, we go over how the results are achieved and give an overview of the results. In section 5 the results are discussed and compared to other papers, finally, in section 6, possible future work is suggested.

2 Methodology

In this section, the focus is on the application of LSTM networks for long-term traffic predictions. First, a brief introduction to LSTMs is provided, followed by a description of the problem and dataset. Lastly, the evaluation metrics for the performance of the LSTM is discussed.

2.1 Understanding Long Short-Term Memory Networks

LSTM models have some advantages over other machine learning models which make them more useful for predicting traffic, regular RNNs are known to suffer from the vanishing gradient problem [10], which hampers their ability to learn from the information in distant parts of the input. LSTM networks overcome this by introducing a memory cell that can maintain information in memory for long periods of time, making them particularly suitable for tasks that involve sequential data with long-term dependencies.

An LSTM unit has three main components: the input gate, the forget gate, and the output gate. These gates collectively determine how much information should be stored or discarded from the cell state at each timestep.

Input Gate: Decides how much of the newly computed information for the current timestep should be stored in the cell.

Forget Gate: Determines how much of the existing information in the cell state should be kept.

Output Gate: Decides what information from the current cell state should be output.

These mechanisms allow LSTM networks to selectively remember or forget things, leading to their ability to handle long-term dependencies in the data.

While LSTM networks have shown their effectiveness in various tasks, it is important to clarify that the objective of this study is not to create the best LSTM model. Rather, the focus is on evaluating the performance of LSTM networks in the context of comparing short and long-term traffic prediction. Looking for the best parameters was not an objective of this study.

The mathematical validity of LSTM networks has been well-established in previous studies, which serve as a solid foundation for the research [11; 12; 13]. These studies also provide a comprehensive understanding of the LSTM architecture, which is why that will not be included in this paper.

2.2 Problem Description

The problem addressed in the research can be summarized as follows: given historical traffic data, the objective is to predict future traffic volumes over different time periods. A formal description of this problem is given by the following:

$$[x_{t-i}, \ldots, x_t] \Rightarrow x_{t+n} \tag{1}$$

Here *i* is the number of timesteps the LSTM looks back to predict the value of the *nth* timestep in the future, and *n* is also the variable that will change during the experiment. x_t is an array of values containing the amount of traffic at that time, depending on how many sensors were used in an experiment the size of this array ranged from 1 to 130 in the experiments.

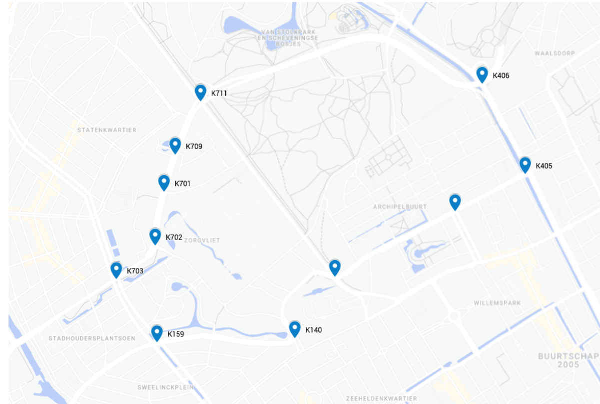


Figure 1: Map showing all sensors

.

Another approach was also considered in which not 1 value was predicted, but all values up to the *nth* value were predicted. This would resolve in the following description:

$$[x_{t-i}, \ldots, x_t] \Rightarrow [x_{t+1}, \ldots, x_{t+n}] \tag{2}$$

After some initial testing, it was concluded that this is not a viable option. It would result in worse predictions and both the x_{t+1}th value and the x_{t+n}th value would be the worst performing resulting in a bowl shape. Especially for the x_{t+1}th value, this was unexpected as it should be the easiest to predict as it is the closest in time.

2.3 Dataset description

The dataset used in this study was collected in November 2019 from detection loops embedded in roads across 11 intersections in The Hague. The dataset comprises readings from a total of 172 sensors, of those 130 contained data for cars and only those were used. The data is aggregated per 15 minutes and shows the number of vehicles passing a sensor during that time period. In total this results in 2880 values per sensor and a total of 374.400 data points. The past 76 timesteps are used to predict a future value, which equals 19 hours, this is also known as the look-back window. The total amount of data points with a value of zero is 35.480, which accounts for 9.5% of all values. Furthermore, there are 1351 missing values represented as *null* in the dataset, accounting for 0.36% of all values.

Figure 1 shows a map of all intersections from which data was collected. The longest distance between 2 sensors was between sensors K703 and K406 and the driving time for that route is 6 minutes according to Google Maps [1]. Because this time is 40% of the time step window, it reduces the spatial dependency between sensors and allows us to focus more on temporal dependencies in the time series. This means that a car measured at a sensor will most like not pass another sensor in the next timestep.

[1]https://www.google.com/maps/

2.4 Evaluation

To evaluate the performance of the LSTM two metrics were used, the Root Mean Square Error (RMSE) and the Symmetric Mean Absolute Percentage Error (SMAPE). The former is also used for the training of the model. For both metrics, the formula is shown for a single sensor, when predicting multiple sensors the average was taken of all sensors.

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(x_t - \hat{x_t})^2}{n}} \tag{3}$$

Equation 3 shows the formula for the RMSE, here $\hat{x_t}$ is the predicted value, x_t is the actual value, and n is the number of values predicted. RMSE is a commonly used metric for regression models and offers an interpretation of how well the model can predict future data. When using the RMSE, larger values will contribute more to the RMSE than smaller values with the same relative error. An error of 10% for values of 2 and 20 will result in an RMSE of relatively 0.2 and 2.

SMAPE and MAPE (Mean Absolute Percentage Error) are commonly used metrics for evaluating forecast accuracy, to compare how close the forecast is to the actual value. Their main difference lies in their calculation of the percentage error. MAPE is computed by taking the absolute difference between the forecasted and actual values, dividing it by the actual value, and multiplying by 100 to obtain the percentage error. However, MAPE's limitation lies in its asymmetry, the following example shows that. Assuming $\hat{x_t}$ is 4 and x_t is equal to 2, will result in a MAPE of 100%, if x_t is 6, it will result in a MAPE of 33%. Assuming both 2 and 6 have an equal chance of being true, lowering the predicted value will result in a better MAPE and thus favor an underprediction.

To address this issue, SMAPE was introduced as an improved version of MAPE. SMAPE takes the average of the forecasted and actual values as a divisor. By using the average of predicted and actual values, SMAPE ensures better treatment of under and over-forecasts. In the previous example, it would result in SMAPE values of 66% and 50%, which are already a lot closer. This approach provides a more balanced measurement of forecast accuracy. Moreover, SMAPE is not affected when the actual value is zero, which can cause division by zero issues in MAPE.

The formulas for MAPE and SMAPE are as follows:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|x_t - \hat{x_t}|}{x_t}\right) \times 100 \tag{4}$$

$$SMAPE = \frac{1}{n}\sum_{t=1}^{n}\left(\frac{|x_t - \hat{x_t}|}{(x_t + \hat{x_t})/2}\right) \times 100 \tag{5}$$

It is worth noting that in cases where the actual value is 0, the calculation of SMAPE is excluded in this research. This exclusion is implemented to avoid distorted results and maintain the meaningfulness of the metric. When the actual value is 0 and the prediction doesn't match, the SMAPE formula would result in a value of 200%, leading to misrepresented results. Excluding these cases ensures a more accurate and realistic assessment of the forecast performance, focusing on situations where comparisons between the forecasted and actual values lead to more insightful values.

While the value of the SMAPE is not relevant at this point, not including the zero values results in the SMAPE going from 40.6% down to 29.1%. This shows that the zero values influence the SMAPE a lot and make a basis for why those values are not included in the calculation.

3 Experimental Setup and Results

This section explains the specific configurations of the experimental setup, how the model was evaluated, and presents the results obtained from the experiments.

3.1 Experimental Setup

The experiments were conducted on a MacBook Pro equipped with an M1 chip, 16GB of RAM, and running macOS 13.5. For the development and execution of the LSTM model, Python 3.9.7 was used along with the TensorFlow 2.12.0 library, utilizing the keras module[2]. This setup is not recommended as Tensorflow is not supported out of the box on newer MacBooks and requires a separate install in order to work with the newer MacBooks[3]. The performance of Tensorflow with the newer MacBooks is also suboptimal and will not work as well as with non-ARM MacBooks or Windows or Linux setups as Tensorflow doesn't use the full capabilities of the MacBook, implied by the following warning given: 'Failed to get CPU frequency: 0 Hz' and the Apple Developer Forum[4] .

For the preprocessing of the data, null values were addressed by forward filling (ffill), which is a method of filling null values in a time series dataset with the previous data point. Additionally, the day and hour features were extracted from the timestamp in the dataset's index and were used as a feature in the prediction, as Zhang and Kabuka [14] have shown improvements using this as a feature.

Lastly, the traffic data was normalized before being fed into the model. Normalization is a common preprocessing step for neural networks, as it scales the data to a smaller range of values. This helps to speed up the training process and can also help to avoid numerical instability issues. In this research, Z-score normalization was used, a method that standardizes the data by removing the mean and scaling to unit variance. This operation involves transforming each feature value into a score that reflects how many standard deviations it is from the mean of the feature.

The LSTM network was constructed with 2 LSTM layers and 1 dense layer, and a batch size of 32 was used for training the model. Although the model was set to train for 500 epochs, Early Stopping callback with a patience of 10 was utilized. This means that if the performance of the LSTM does not improve over the last 10 epochs, it will stop training. This mechanism allows the model to stop training when it no longer improves, saving computational resources and preventing overfitting. The learning rate was set to 0.001, and the Adam optimizer was used during the learning process.

The dataset was split into 22 days for training, 3 days for validation, and 5,2 days for testing. That results in 22 days

[2]https://www.tensorflow.org/

[3]https://developer.apple.com/metal/tensorflow-plugin/

[4]https://developer.apple.com/forums

which the LSTM uses to learn patterns in the data and 3 days to validate while training what the RMSE is. The last 4 and 5 hours days are used after training to see how it performs on unknown data and is also the RMSE used for the results. The 19 hours that are missing from the data, are the first 19 hours of the dataset, they are only used for labeling the data.

3.2 Performance Evaluation Method

The primary focus of this research was on evaluating the model's capability to predict traffic volume at different future time steps, where each step corresponds to 15-minute intervals. To achieve this, a time-lagged labeling technique was used, where the label for a given instance was the traffic volume at a future time step *i*. For example, if *i = 1*, the model was tasked with predicting the traffic volume 15 minutes into the future; if *i = 5*, the model predicted the volume 75 minutes ahead, and so on.

The first 5 timesteps were all evaluated, and after those results, timesteps were evaluated in a stepwise manner advancing by one hour (i.e., four 15-minute steps) at a time. In this manner, a wide range of future time steps were able to be covered, starting from *i = 1* (15 minutes into the future) up to *i = 41* (10 hours and 15 minutes into the future).

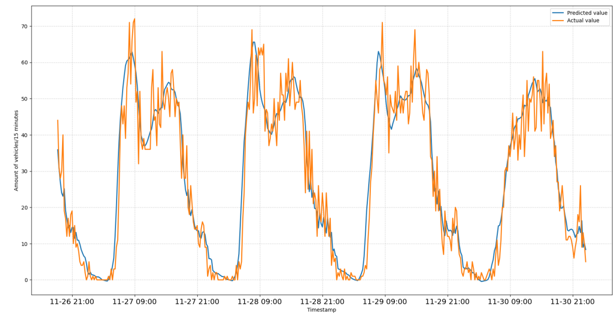
To ensure a robust evaluation and mitigate the effects of randomness or noise created by the initial conditions of the LSTM, for each time step *i*, 20 individual runs of the LSTM were performed. The final predicted value for each time step was then computed as the average of these 20 runs. This approach provided a more reliable estimate of the LSTM model's performance at each time step.

3.3 Results

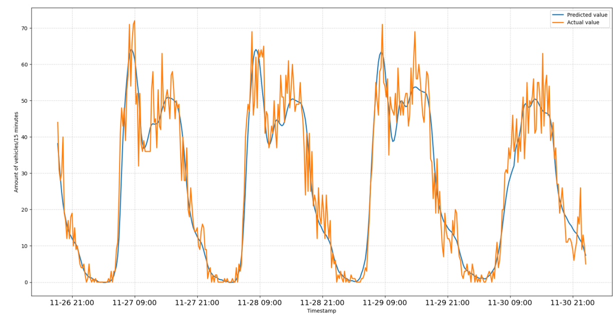
Figure 2a shows the results when predicting values for a single sensor 1 timestep in the future. What is apparent about the values is that they can fluctuate a lot between 2 timesteps, at *11-29 09:15* for example the amount of cars is 56, with the next value being 35 and the following 51. These are differences of 38% and 51% respectively. It can be seen in the graph that the LSTM is unable to pick up these high fluctuations accurately, this is for example visible at *11-27 9:00*. The actual traffic is at 70 cars with the previous value being 54 cars, while the LSTM predicts 62 cars. The choice was made to show the prediction for a single sensor and not the average of predicting all 130 sensors individually. Taking all sensors would average out the fluctuations and give a graph that doesn't show that for sequential timesteps the values can differ a lot. The trend of the sensor chosen is similar to that of the average of all sensors.

Figure 2b shows the predictions for 41 timesteps in the future for the same sensor as in 2a. When comparing the results between a single timestep and those of 41 timesteps in the future it is visible that spikes in traffic are not able to be picked up. This is apparent at the *11-30 21:00* timestamp, the prediction follows a smooth curve, while the single timestep LSTM can pick up that there is a lower amount of traffic than usual.

Table 1 shows the change of the RMSE and SMAPE for different timesteps for the same sensor as in figure 2. For the RMSE a steady increase is visible throughout extending the



(a) Predicting 1 timestep in the future, with 1 sensor.



(b) Predicting 41 timesteps in the future, with 1 sensor.

Figure 2: Comparison between 1 and 41 timesteps for 1 sensor.

prediction horizon. However, for the SMAPE nothing can be concluded, it differs a lot over the different timesteps.

Figure 3a presents the average of all predictions for 130 sensors. This shows that over the average of all values, there are a lot fewer fluctuations in traffic, as these fluctuations are averaged out against each other. While from these graphs it looks like the predictions are almost perfect, for each individual sensor the results are not perfect.

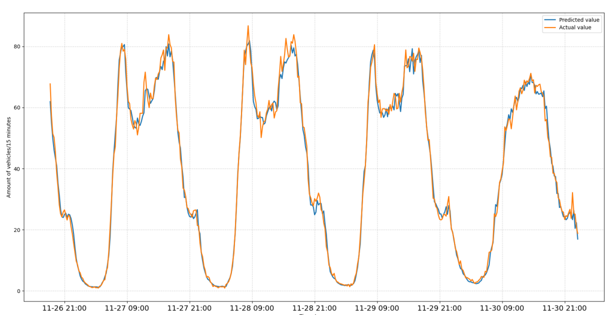
Comparing the results from 2b with the LSTM predicting 41 timesteps in the future as shown in 3b, the results are comparable to the single sensor predictions. Sudden spikes in traffic are not able to be predicted as visible around the *11-27 11:00* mark, when predicting a single timestep the LSTM is able to pick this up.

In table 2 the results are shown when predicting values from a single timestep up to 41 timesteps in the future for all sensors, including the percentual change compared to the single timestep prediction. The values in the table are rounded after calculation, which is why the percentage differences do not match exactly when comparing them to the values in the table. What is noticeable is that the performance degrades quickly from a single timestep to 5 timesteps. The RMSE increases by 6.5% and the SMAPE by 5.6%. The next 6% is only visible at the 21st timestep for the RMSE, while for the SMAPE the increase at that timestep has decreased by 7.8% compared to a single timestep prediction. Surprisingly, the SMAPE starts to drop after the 25th timestep and is lower than the baseline at the 41st timestep.

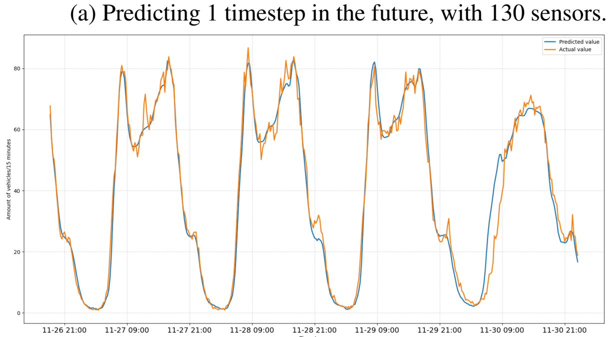
In figure 4 the performance of both the RMSE and SMAPE can be seen in comparison. Here it is easily visible how the RMSE keeps increasing, while thee SMAPE starts to de-

Table 1: Rounded evaluation Metrics of LSTM Model for different timesteps, showing percentual change compared to 15 minutes for 1 sensor

Hours	RMSE	Change	SMAPE %	Change
0.25	5.8	–	24.6	–
0.5	5.9	1.4%	24.6	0.0%
0.75	6.0	2.9%	26.2	6.8%
1.0	6.1	4.7%	24.8	1.0%
1.25	6.0	4.0%	25.6	4.2%
2.25	6.0	3.2%	25.1	2.2%
3.25	6.1	4.9%	24.9	1.5%
4.25	6.0	3.7%	24.5	-0.4%
5.25	6.0	3.5%	24.3	-1.1%
6.25	6.1	4.9%	25.2	2.6%
7.25	6.3	8.3%	25.7	4.8%
8.25	6.2	7.6%	27.7	12.9%
9.25	6.2	6.6%	26.0	6.0%
10.25	6.4	10.4%	25.0	1.9%



(a) Predicting 1 timestep in the future, with 130 sensors.



(b) Predicting 41 timesteps in the future, with 130 sensors.

Figure 3: Comparison between 1 and 41 timesteps for all sensors.

Table 2: Rounded evaluation Metrics of LSTM Model for different timesteps, showing percentual change compared to 15 minutes for all sensors

Hours	RMSE	Change	SMAPE %	Change
0.25	7.6	–	28.8	–
0.5	7.8	2.1%	29.5	2.4%
0.75	7.9	4.0%	29.9	3.7%
1.0	8.0	5.0%	30.3	5.2%
1.25	8.1	6.5%	30.4	5.6%
2.25	8.1	6.4%	30.0	4.0%
3.25	8.4	10.6%	30.2	4.9%
4.25	8.4	10.6%	31.1	7.8%
5.25	8.5	12.3%	31.1	7.9%
6.25	8.6	12.7%	32.3	12.2%
7.25	8.7	15.2%	31.9	10.7%
8.25	9.0	18.9%	30.7	6.6%
9.25	8.9	18.9%	29.2	1.5%
10.25	9.0	19.1%	27.7	-3.8%

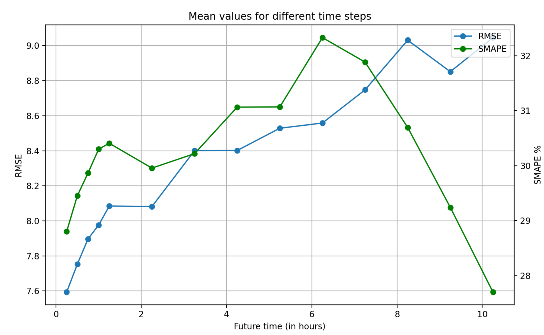


Figure 4: RMSE and SMAPE comparison per timestep

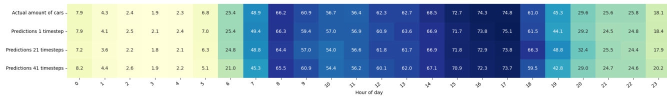
.

crease after 6 hours.

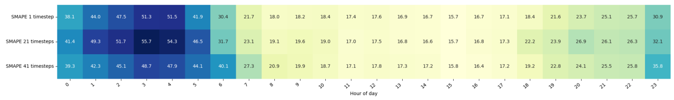
In figure 5 an overview is visible of the average prediction of cars per hour, the SMAPE per hour, and the RMSE per hour for 3 different timesteps for the test dataset. A lighter color means a lower value, while the darker the cell gets the higher the value compared to the rest of the heatmap. Do note that due to the experimental setup, these values are not the same as in table 2 and 4, but still show the same differences with each timestep.

Figure 5a shows the actual traffic for each hour in comparison to the different timesteps. On average if the prediction horizon gets bigger, the predicted values will get lower. This is also visible in table 3, which shows the average prediction over the complete day and the average of the percentual difference for each hour.

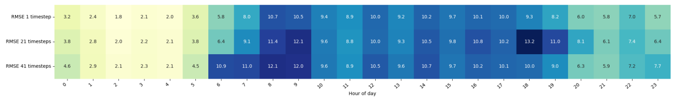
In figure 5b the SMAPE is visible per hour of the day. What can be seen is that during the night, when traffic is low, the SMAPE is twice or sometimes three times as high compared to during the day. An explanation for this could be that an absolute error of 1 or 2 will have more effect when the values are low than when the values are high. For example, an error of +2 with the actual values being 1 or 20 will result



(a) Actual traffic and predictions per hour of the day



(b) SMAPE % per hour of the day



(c) RMSE per hour of the day

Figure 5: SMAPE and RMSE comparison per hour for different timesteps.

in a SMAPE of respectively 50% and 9.5%. During the night big relative differences are also seen, just like during the day, an example is a value dropping from 11 to 4, a 64% decrease. The LSTM predicted around 14 cars for this sensor resulting in a SMAPE of 111%. It is also evident from this heatmap that the 21st timestep predictions are worse than the other 2 during the night.

For the RMSE the biggest difference lies in the morning peak, where the difference is sometimes almost doubled for the 41 timesteps in comparison to the single timestep as seen in 5c. Also, the afternoon peak is worse for the 21 timestep prediction and a little worse for the 41 timestep prediction.

Table 3: Comparison of average predictions against actual values

Average	Actual	1 timestep	21 timesteps	41 timesteps
Amount of cars	40.0	39.8	39.7	38.9
Difference per hour	–	-0.09%	-2.5%	-2.8%

4 Responsible Research

Ethical issues should be taken into account when doing research with data, in the case of this research there was 1 possibility for an ethical issue which is the identifiability of a specific person with the help of the data. Since the data was aggregated by 15 minutes it is anonymized. Since this was the only possibility for an unethical point in the research done, there are no further ethical issues that need to be addressed.

5 Discussion and conclusion

With an RMSE of 7.6 and SMAPE of 28.8% for the all-sensor model for a single timestep prediction, the RMSE is in line with other research done, although the SMAPE is higher than other papers [15; 16]. The results discussed are about the all-sensor model, as the results for a single sensor cannot be representative for all sensors.

However, it is important to note that the effectiveness of LSTM models can vary depending on the context and dataset size. For instance, an RMSE of 6.75 was achieved in [17], which is lower than the result in this paper. But, they were working with a dataset three times larger, offering a wider base for the model to learn from, which could contribute to their lower error rate. They reported a MAPE of 17.14%, which is lower than the SMAPE in this paper of 28.8%. However, their data was from highway traffic, typically registering higher traffic volumes. As shown earlier working with smaller values can lead to a higher SMAPE [4].

As the prediction horizon extended from 15 minutes to 75 minutes, a noticeable degradation in the LSTM model's performance was observed for both metrics. Yet, the RMSE increase was less steep after predicting further than 75 minutes compared to the initial increase. Additionally, the SMAPE increased up to 6 hours, but then showed improvements, surpassing the accuracy of predictions for 15 minutes into the future after 10 hours. The results also show that as the time horizon increases the LSTM tends to under forecast the traffic on average.

For the RMSE the hours showing the worst performance were the morning and afternoon peak. Especially the morning rush became worse as the prediction horizon grew, while the night hours had almost no contribution to the RMSE. The SMAPE was however highly impacted by the low values in the night, showing values of up to three times as high as during the day. Overall, for this dataset, the LSTM will predict slightly lower values as the prediction horizon grows.

6 Future Work

Looking ahead, there are several potential paths for future work in this area. Firstly, having a larger dataset could potentially improve the performance of the LSTM model, as observed in the lower error rates reported by Yu et al. (2017)[17], who used a dataset three times larger than ours.

Furthermore, the granularity of the data could impact the results achieved. Often the interval is between 5 and 10 minutes [2; 3], while the data used in this research was aggregated by 15 minutes. Decreasing the interval could also open the way to use spatial dependencies between sensors.

Thirdly, incorporating external factors, such as weather data or event data, could enhance the model's ability to anticipate changes in traffic flow. This could improve the challenge of predicting traffic spikes, as [14] showed a performance increase when incorporating different external factors.

Finally, exploring other machine learning approaches or refining the current LSTM model may help improve the accuracy of long-term forecasts as the perfect LSTM parameters were not the goal of this research.

References

[1] H. Liu. *Travel Time Prediction for Urban Networks*. PhD thesis, TU Delft, 10 2008.

[2] Kian Lun Soon, Robin Kuok Cheong Chan, Joanne Mun-Yee Lim, and Rajendran Parthiban. Short-term traffic forecasting model – prevailing trends and guidelines. *Transportation Safety and Environment*, 12 2022. tdac058.

[3] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, 43:3–19, 2014. Special Issue on Short-term Traffic Flow Forecasting.

[4] Licheng Qu, Wei Li, Wenjing Li, Dongfang Ma, and Yinhai Wang. Daily long-term traffic flow forecasting based on a deep neural network. *Expert Systems with Applications*, 121:304–312, 2019.

[5] Xingyi Li Zhongsheng Hou. Repeatability and similarity of freeway traffic flow and long-term prediction under big data. *IEEE Transactions on Intelligent Transportation Systems*, 17(6):1786–1796, 2016.

[6] Jinbao Li Yan Wang, Qianqian Ren. Spatial–temporal multi-feature fusion network for long short-term traffic prediction. *Expert Systems with Applications*, 224:119959, 2023.

[7] Rongzhou Huang, Chuyin Huang, Yubao Liu, Genan Dai, and Weiyang Kong. Lsgcn: Long short-term traffic prediction with graph convolutional networks. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 2355–2361. International Joint Conferences on Artificial Intelligence Organization, 7 2020. Main track.

[8] Yuankai Wu, Huachun Tan, Lingqiao Qin, Bin Ran, and Zhuxi Jiang. A hybrid deep learning based traffic flow prediction method and its understanding. *Transportation Research Part C: Emerging Technologies*, 90:166–180, 2018.

[9] Jiayu Liu, Xingju Wang, Yanting Li, Xuejian Kang, and Lu Gao. Method of evaluating and predicting traffic state of highway network based on deep learning. *Journal of Advanced Transportation*, 2021:8878494, 2021.

[10] Sepp Hochreiter. The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 06(02):107–116, 1998.

[11] Frederik Kratzert, Daniel Klotz, Claire Brenner, Karsten Schulz, and Mathew Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018.

[12] Venkat Venkatasubramanian. The promise of artificial intelligence in chemical engineering: Is it here, finally? *AIChE Journal*, 65(2):466–478, 2019.

[13] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.

[14] Mansur R. Kabuka Da Zhang. Combining weather condition data to predict traffic flow: A gru based deep learning approach. *IET Intelligent Transport Systems*, 12, 03 2018.

[15] Mingfang Huang Jianhu Zheng. Traffic flow forecast through time series analysis based on deep learning. *IEEE Access*, 8:82562–82570, 2020.

[16] Toon Bogaerts, Antonio D. Masegosa, Juan S. Angarita-Zapata, Enrique Onieva, and Peter Hellinckx. A graph cnn-lstm neural network for short and long-term traffic forecasting based on trajectory data. *Transportation Research Part C: Emerging Technologies*, 112:62–77, 2020.

[17] Haiyang Yu, Zhihai Wu, Shuqin Wang, Yunpeng Wang, and Xiaolei Ma. Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors*, 17(7), 2017.