Assessing Methods for Handling Missing Data Using an LSTM Deep Learning Model in Traffic Forecasting

Wouter Büthker[1]

Supervisor: Elena Congeduti[1]

[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: Wouter Büthker
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, George Iosifidis

Abstract

Due to the increasing popularity of various types of sensors in traffic management, it has become significantly easier to collect data on traffic flow. However, the integrity of these data sets is often compromised due to missing values resulting from sensor failures, communication errors, and other malfunctions. This study investigates the effect of missing data on the performance of Long Short-Term Memory (LSTM) models in traffic flow prediction and assesses strategies to handle these missing values. By actively removing values from a complete data set, three strategies to handle these missing values are evaluated: dropping null values, replacing them with zero, and linear interpolation. We show that LSTM models are surprisingly resilient to missing data, with little impact on prediction accuracy for up to 40% missing data, irrespective of the strategy used. For higher proportions of missing data, dropping null values leads to significant performance degradation, while zero-filling and interpolation maintain predictive accuracy. This paper provides insights into the choice of missing data handling strategies in time-series prediction tasks, demonstrating the potential of LSTM models for traffic forecasting under less-than-ideal data conditions.

1 Introduction

In the modern economic environment, growth is everything. As the economy and cities grow, so do the number of vehicles and the demand for infrastructure to accommodate these cars. Accurate traffic prediction can improve transportation efficiency, reduce traffic congestion, and create safer roads [3]. Using good predictions the traffic network can be used to its fullest extent. This can reduce the need for incredibly expensive infrastructure expansions. Moreover, traffic prediction is a critical component of Intelligent Transportation Systems (ITS), which are becoming increasingly prevalent in modern cities. Optimizing predictions can, among other things lead to more efficient traffic light management. By understanding the significance of traffic prediction, better transportation systems that are efficient, sustainable, and safe for everyone can be developed.

There are multiple ways to collect data on traffic flow, where traffic flow is defined as the total number of cars passing a given point in a given time. In this study, data from the municipality of The Hague, The Netherlands is used. Traffic flow is detected by using inner-city induction loops. Induction loops are usually installed at traffic lights and can detect both the amount of cars that passed the sensor, and for how long they were on top of the sensor.

Recent research into traffic forecasting has indicated a shift towards the use of Long Short-Term Memory (LSTM) [4] models due to their ability to capture complex temporal patterns, an aspect integral to traffic flow data. Studies such as [11] and [1] showed promising results in using LSTM for short-term traffic flow prediction, outperforming traditional time-series models and shallow neural networks [7].

LSTM models are particularly beneficial in this domain due to their capability to remember long-term dependencies, which is crucial considering the temporal dependencies (time of day, day of week, season) and irregularities (accidents, construction work) inherent in traffic flow data. Moreover, LSTM's gate mechanisms help to avoid the problem of vanishing and exploding gradients, enabling the model to learn effectively over many time steps, a significant advantage for reliable traffic forecasting.

Unfortunately, real-world data often contain inaccuracies. This is not different in the traffic domain, broken sensors or problems in signal transmission can result in gaps in the data. In some cases, broken sensors can also result in artificial detections. The inaccuracies can possibly heavily compromise prediction accuracy. Often these inaccurate data points are removed to train the model, resulting in smaller training set sizes and possibly inaccurate results using real-world data. However, there are some approaches that try to prevent missing data in the first place by imputing missing values in the traffic domain [2] or in other domains where LSTMs are used [6][5]. By proper understanding of missing data and ways to mitigate model performance decreases on incomplete data sets, better performance on real-world data can be achieved.

This paper aims to answer the following questions:

1. How much does missing data affect the accuracy of an LSTM traffic prediction model?
2. What is an effective strategy to improve model performance using erroneous data?

A deep learning LSTM model based on [10], [1], and [11] is used to evaluate the current traffic volumes and the possible impact of inaccurate data. Inaccurate data can consist of missing, but also of erroneous data, due to for instance broken sensors.

Firstly, the created LSTM model is described in Section 2.1. Subsequently, detecting and marking this inaccurate data is discussed in Section 2.2, based on raw and aggregated data. Section 2.3 describes the methodology of research. Furthermore, Section 3.1 discusses the experimental setup in detail. Afterwards, in Section 3.2 the effects of missing data in the prediction accuracy of the model are evaluated. In Section 4 some ethical concerns are reviewed. Finally, Section 5 concludes and Section 6 describes further work.

2 Methodology and Techniques

This section discusses the LSTM in Subsection 2.1. Subsequently, the source data is analyzed in detail in Subsection 2.2. Finally, the problem is formalized and the basic experimental setup is explained in Subsection 2.3.

2.1 Long Short-Term Memory

To evaluate the effects of incorrect data on traffic predictions, a Long Short-Term Memory (LSTM) model was created to execute the required experiments.

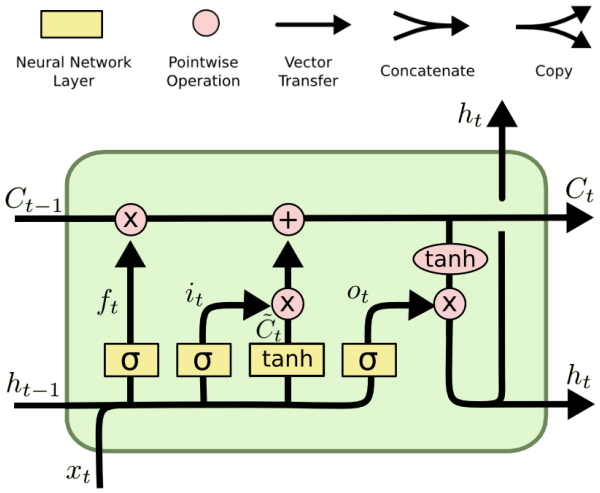

Figure 1: Architecture of a single LSTM module. Adapted from [8].

LSTMs have shown to be effective at sequence prediction. The problems other Recurrent Neural Networks (RNN) encounter in sequence prediction such as difficulty to learn long-term dependencies result from the exploding/vanishing gradient problem. This problem occurs when the gradient is propagated over many layers. LSTMs use both a short- and long-term memory to overcome this problem.

A standard LSTM unit is composed of a cell, an input gate, an output gate, and a forget gate. The cell remembers values over arbitrary time intervals, and the three gates regulate the flow of information into and out of the cell. An illustration of an LSTM module adapted from [8] can be found in Figure 1.

Formally, an LSTM cell at a time step t in its simplest form is defined as follows:

First, we compute the forget gate f_t:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \tag{1}$$

where W_f are the weights for the forget gate, h_{t-1} is the hidden state from the previous time step, x_t is the input at the current time step, b_f is the bias for the forget gate, and σ is the $sigmoid$ function.

The input gate i_t and the candidate cell state $\tilde{C}_t$ are computed as follows:

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

Here, W_i and b_i are the weights and bias for the input gate, and W_C and b_C are the weights and bias for creating the new candidate cell state. $tanh$ is the hyperbolic tangent function, which outputs values between -1 and 1.

The new cell state C_t is computed as:

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{C}_t \tag{4}$$

Here, $\circ$ denotes element-wise multiplication. The old cell state C_{t-1} is forgotten according to f_t and incremented by the candidate cell state $\tilde{C}_t$ scaled by the input gate i_t.

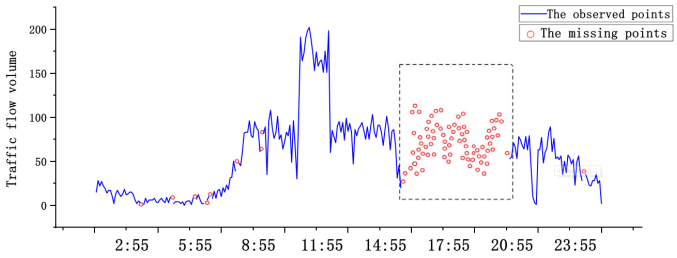

Figure 2: Patterns of missing data. From [10].

The output gate o_t is computed as:

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

Where W_o and b_o are the weights and bias for the output gate.

Finally, the new hidden state h_t is computed as:

$$h_t = o_t \circ tanh(C_t) \tag{6}$$

The $tanh$ of the cell state C_t is output, but only scaled by the output gate.

In summary, an LSTM is able to selectively forget its cell state, selectively update its cell state, and selectively output its cell state, due to the gating mechanisms provided by the forget gate, input gate, and output gate, respectively. This makes LSTMs useful for tasks where longer time dependencies are required, like in some sequence prediction tasks.

2.2 Preliminary Data Analysis

The data set used in this paper contains data from the municipality of The Hague. Induction loops, often in front of traffic lights, were used to detect cars, bicycles, and trams. The raw data set contains a total of 172 sensors, divided over 11 intersections. Data was collected in the month of November 2019 and stored in the V-LOG [9] data format. In this format, the start and end time of a sensor detecting a car is stored to create a timeline of all detections per sensor.

To use this data as input for the LSTM, multiple preprocessing steps have been adopted. Data from bicycles and trams was removed as the focus of this research is on car traffic. Data from the timeline was aggregated into a simple count of cars per 15 minutes, the traffic flow. This flow is then normalized per sensor using the standard score $z = (x - u)/s$, where x is a sample, u is the mean of samples and s is the standard deviation.

The data set contains 2880 time steps of 15 minutes for 130 sensors. In total, there are 1351 missing values in this whole data set. The missing values appear as $null$ values in the source data. This makes 0.36% off all values $null$ values. Some sensors contain more erroneous data than others. The low amount of missing data in this data set allows for insights into how much missing data will have an impact on prediction accuracy.

After simple statistical analysis, it can be determined $null$ values in the source data appear in two patterns:

1. In the first pattern, single values are missing at often regular intervals in the data. Conversion errors seem to be

the main cause for this, based on some manual comparisons with the raw data format (V-LOG). In other cases, the missing values are spread out randomly throughout the data. These errors could be caused by hardware failure or problems in signal transmission.

2. Whole uninterrupted sequences missing from the data is the second possible pattern. Only if there are more than three consecutive values missing, it is considered a missing sequence. An example of a missing sequence is outlined in Figure 2. Based on the source data, missing sequences have a length of 9.8 time steps on average, with a standard deviation of 3.3. These kinds of errors are more likely due to hardware failure.

In this research, the focus will be on the second pattern, as this pattern appears most often, and imputing a single missing value can be trivial.

2.3 Methodology

To answer the research questions specified in Section 1, the LSTM described in Section 2.1 is used to complete the task of traffic flow prediction. In short, an artificial data set is created by removing values from the original temporal sequence. This partially destroyed data set is then used to compare different strategies for handling missing data.

Formally, Given a set of historical traffic flow observations $S = (x_1, x_2, ..., x_n)$ where x_i denotes the traffic flow at time step i, we create a subsequence $X = (x_s, x_{s+1}, ..., x_t)$, where $X \subset S$. The aim is to predict the traffic flow y_{t+1} at a future time step $t + 1$.

The model is trained by minimizing the loss function for the Root Mean Squared Error (RMSE), which measures the discrepancy between the predicted traffic flow and the true traffic flow. The true traffic flow at time $t + 1$ is denoted by y_{t+1}. When data has not been manipulated from the training set $y_{t+1} = x_{t+1}$. The total amount of sequences is denoted by n. With this, the loss function can be written as:

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

To determine the impact of missing data and possible ways to mitigate the impact, multiple strategies for handling missing data are investigated.

1. The first strategy is dropping all $null$ values. This would create a smaller data set $S'_p = (x_1, x_2, ..., x_{t-p})$, where p is the amount of $null$ values. The values might be removed at multiple different time steps in S'_p. This results in a set where the assumption that time steps have regular intervals is voided.
2. Another strategy is to replace all $null$ values with 0. From this we get $S'_p = (x_1, x_2, ..., x_t)$ where p values of x have been replaced with 0.
3. The final strategy to be explored is using linear interpolation $S'_p = (x_1, x_2, ..., x_t)$ is created where p values of x_i have been interpolated using Formula 8, where x_l is the closest non-null value preceding x_i and x_r is the closest non-null value following x_i.

$$\tilde{x}_i = x_l + \frac{x_r - x_l}{r - l} \cdot (i - l) \tag{8}$$

Regardless of the strategy, the model is trained on subsequences $(X_1, ..., X_n)$ based on the artificial data set S'. Finally, the model is evaluated against a complete baseline data set S. Multiple different values for p are used to determine how the amount of missing data affects each strategy. This experiment is then repeated for multiple sensors to provide more reliable results. Average measures over the different sensors are considered to assess the effectiveness of the different strategies.

3 Experimental Setup and Results

This Section firstly discusses the exact experiment details in 3.1. Subsequently, the results and findings are discussed in Subsection 3.2.

3.1 Experimental Setup and Data Preparation

Firstly, the effect of missing data has to be detected. The baseline for the experiments should be a complete data set without any missing values. The original data set contains only a small set (0.36%) of missing values.

By manual data selection, 84 of the 130 sensors that do not have any missing sequences are used. These sensors still contain some sparse missing values, but never in sequences of more than three. To be able to train the model the data set cannot contain any missing values. Therefore, imputation is used to fill these values and create a baseline data set.

While imputation on the baseline data set should generally be avoided, since there are only a small number of values missing and a maximum of three values in a row are missing, it is assumed linear interpolation would not cause impactful inaccuracies.

From the baseline data set, the data is divided into subsequences using a sliding window. The traffic flow values are put into sequences of 80 steps (20 hours), using the last value as a label. These sequences are then split into a training set (60%), a validation set (15%), and a test set (25%). The validation and test set are not affected by data removal and missing data strategies.

The data is then removed as described in Section 2.3 using processes and choices that are consistent with the missing data occurrences observed in our original data set as reported in Section 2.2.

Different methods to remove data from the baseline data set have been considered. A fixed sequence length of 10 was chosen, based on the observations described in Section 2.2. To remove sequences of data, random indices are chosen from where a sequence is removed under the constraint that sequences never overlap. The performance of the learning model is tested for different amounts of manipulated sequences.

Using this new broken training set, multiple strategies to handle the $null$ values are utilized as described in Section 2.3. Using the first strategy, $null$ values are simply dropped. In the second strategy, values are replaced by 0. This is done after normalizing, as a result, the values are in reality being

Figure 3: LSTM performance on recomputed data sets

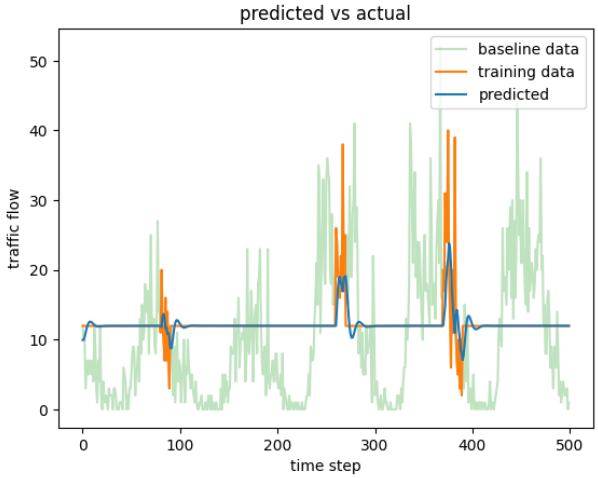

Figure 4: Model performance on the training set with 95% percent of data removed

replaced by the mean traffic flow. Linear interpolation is done according to Formula 8. To recompute values at the edges, not in between two non-null values, the outer-most value is simply repeated.

Using the different data sets the model is trained and evaluated against the baseline data set.

The model consists of a total of four layers: an input layer, two LSTM layers, and a dense output layer. The LSTM layers both contain 60 units. The model is trained for a maximum of 100 epochs, with an early stopping condition if performance on the validation set does not increase for 10 epochs. As mentioned before RMSE, is used as the loss function. The Adam optimizer is used with a learning rate of 10E-5.

Experiments are run for regular intervals of 5% of missing data between 0% and 95% (inclusive) for a total of 20 runs per strategy. The experiments are repeated for multiple sensors and results are averaged per strategy to ensure reliability and validity. A total of 5 random sensors from the baseline data set were used.

3.2 Results and Findings

This Section describes the results and findings of the experiments described in Section 2.3 and Section 3.1.

Figure 3 clearly illustrates that the RMSE increases as larger proportions of data are removed, indicating a decline in model performance. Remarkably, the model seems resilient to missing data up to 40%, regardless of the strategy used to handle missing data. Despite this, the model still yields reasonably accurate predictions even with substantial data removal.

The strategy of dropping missing values underperforms when more than 40% of data is removed, and the performance further decreases when the missing data exceeds 75%. Interestingly, there are instances where the RMSE decreases even though more data is removed. This counterintuitive result may suggest the existence of noise or non-informative data in the removed sections, although further research is needed to understand this.

The strategies of setting missing values to zero and interpolation maintain effective performance, even with a significant percentage of data missing. The set-to-zero strategy only results in a 26% increase in RMSE even when 95% of the original data is missing.

Figure 4 provides an example of a prediction of the traffic flow at a randomly picked sensor for approximately five days. The actual traffic flow is shown as baseline data. 95% of this data is destroyed and replaced with 0. Note that this process is done while the data is normalized, resulting in the 0 values being replaced by the mean traffic flow when the data is unnormalized. The data recomputed using the set-to-zero strategy is denoted as training data. For this example, you can see only three sequences have not been destroyed and overlap with the baseline data. Finally, the model's predictions are shown, trying to mimic the training data.

To further investigate the surprisingly good performance of a learning model trained with a major proportion of broken sequences, Figures 4 and 5 depict the predictions over 5 days over the manipulated training set.

Figure 5 shows the prediction over a test set of the same model trained using only 5% of original data, but instead on a test set of complete data. Surprisingly, the model is still able to capture the variations when original sequences of data are presented to the forecasting model. However, it is visible that on traffic flow peaks the model predicts too low values, while on dips it predicts too high values.

Figure 6 shows a model trained on the complete baseline data set. In this data set, no data has been removed. The model predicts on the same test set as in Figure 5. Compared to Figure 5 it is visible that Figure 6 has a more smooth prediction curve and predicts the actual values a little better. This is especially clear at dips when the actual traffic flow hits zero, during quiet times at night. At peaks, during rush hour, the model manages to predict the average traffic flow better too.

The model trained on only 5% of data has only a minimal decrease in performance compared to the baseline model.

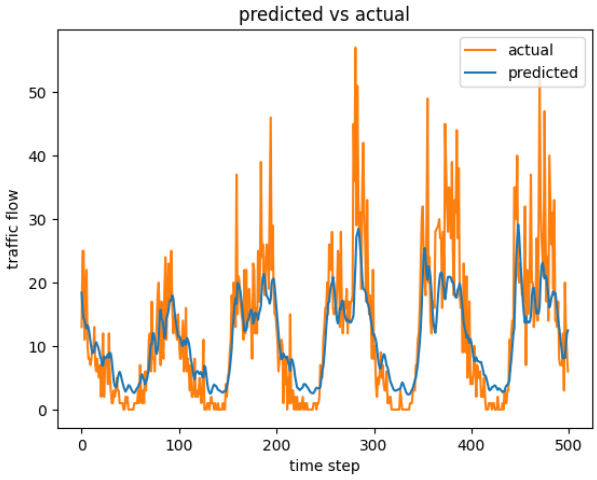

Figure 5: Model performance on the test set when trained with 95% of data removed

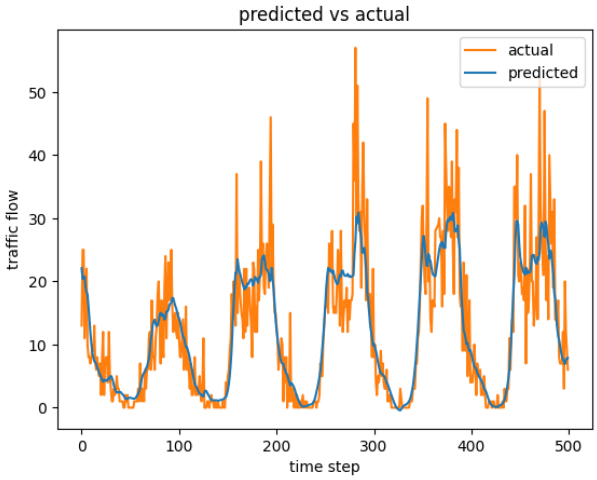

Figure 6: Model performance on the test set when trained on the baseline data set

Figure 4 and 5 give an intuition of why the performance measures remain good even with little data. While the model has more preference to stay near the traffic flow mean, it exhibits a remarkable ability to learn patterns from the limited data and predict trends accurately. Its conservative approach serves as a stabilizing force, minimizing the chance of drastic over-estimations or under-estimations.

4 Responsible Research

In conducting this study, ethical standards in data usage and experimental replication are followed. The traffic flow data used in the study is collected from induction loop sensors, ensuring anonymity and privacy. This type of traffic data only captures traffic flow information without any identifiers, maintaining the privacy of individual road users. Therefore, it is a non-intrusive and ethically compliant source of data for our traffic prediction task.

Furthermore, to ensure the reliability and validity of our findings, we have undertaken multiple runs of our experiments. This approach reduces the likelihood of random variations or outliers influencing our results. This approach ensures the trustworthiness and reproducibility of our research findings, contributing to responsible and robust research practices.

5 Conclusions and Discussion

Based on the results it can be concluded that surprisingly, even for high amounts of missing data the model is still able to make fairly good predictions.

For less than 40% of the data missing, the choice of strategy to handle missing data does not have much impact. In the case where there is more data missing, dropping values should be avoided, since it proved to be less reliable than the other strategies. Both interpolation and setting values to zero lead to a small predictable increase in RMSE.

Although the performance of the model understandably declines with higher rates of missing data, certain strategies such as zero-filling and interpolation are remarkably effective in maintaining model performance. This study thus provides valuable insights into the choice of strategies for handling missing data in time-series prediction tasks, demonstrating the potential of LSTM models for traffic forecasting even in less-than-ideal data conditions.

6 Further work

An important topic to be explored further is model performance on missing data when the model has been trained with a perfect data set. A similar process as described in this paper could be used, with the exception that data is removed (and recomputed) from the test set instead of the training set. Based on this informed choices in strategies for handling missing data in real-world scenarios can be made.

Besides missing data, broken sensors can also generate excess data that might not represent real-world scenarios. Further work could look into how these kinds of errors can be detected, how they affect a model, and how detrimental effects could be mitigated.

This research focused on strategies to handle sequences of missing values. However, based on the source data analysis in Section 2.2, this is not the only pattern in which data is missing. Single missing values at regular intervals or random locations are also present. The effect of missing values following this pattern on a model should be explored as well.

References

[1] Zainab Abbas, Ahmad Al-Shishtawy, Sarunas Girdzijauskas, and Vladimir Vlassov. Short-term traffic prediction using long short-term memory neural networks. In *2018 IEEE International Congress on Big Data (BigData Congress)*, pages 57–65, 2018.

[2] Yanjie Duan, Yisheng Lv, Wenwen Kang, and Yifei Zhao. A deep learning based approach for traffic data imputation. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 912–917, 2014.

[3] S. Foo and B. Abdulhai. Evaluating the impacts of changeable message signs on traffic diversion. In *2006 IEEE Intelligent Transportation Systems Conference*, pages 891–896, 2006.

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

[5] Sunghoon Lim, Sun Jun Kim, YoungJae Park, and Nahyun Kwon. A deep learning-based time series model with missing value handling techniques to predict various types of liquid cargo traffic. *Expert Systems with Applications*, 184:115532, 2021.

[6] Xinxi Lu, Lijuan Yuan, Ruifeng Li, Zhihuan Xing, Ning Yao, and Yichun Yu. An improved bi-lstm-based missing value imputation approach for pregnancy examination data. *Algorithms*, 16(1), 2023.

[7] Yisheng Lv, Yanjie Duan, Wenwen Kang, Zhengxi Li, and Fei-Yue Wang. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):865–873, 2015.

[8] Christopher Olah. Understanding lstm networks, Aug 2015.

[9] Claassens Solutions. V-log protocol en definities, July 2020.

[10] Yan Tian, Kaili Zhang, Jianyuan Li, Xianxuan Lin, and Bailin Yang. Lstm-based traffic flow prediction with missing data. *Neurocomputing*, 318:297–305, 2018.

[11] Zheng Zhao, Weihai Chen, Xingming Wu, Peter C. Y. Chen, and Jingmeng Liu. Lstm network: a deep learning approach for short-term traffic forecast. *IET Intelligent Transport Systems*, 11(2):68–75, 2017.