Graph Neural Networks Training Set Analysis

Effect of Training Data Size

Alexandru-Valer Păcurar[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 23, 2024

Name of the student: Alexandru-Valer Păcurar
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, Lilika Markatou

Abstract

With the rapid increase in popularity of graph neural networks (GNNs) for the task of traffic forecasting, understanding the inner workings of these complex models becomes more important. This experiment aims to deepen our understanding of the importance that the training data has in regards to the ability of GNNs to accurately predict traffic. By repeatedly training the same GNN model with different training datasets spanning over various time frames and comparing standard performance metrics computed based on the predictions performed by the model, this paper concludes that while using less training data leads to a slight decrease in performance, this is heavily dependent on the quality of the dataset. If the data gathering process is short and the sensors are not properly maintained, GNNs are not able to accurately predict traffic. On the other hand, if the data gathering process goes well and there are few missing values, GNNs perform well even when trained with smaller amounts of historical data.

Key words: GNN, traffic forecasting, training data

1 Introduction

In the evolving landscape of urban mobility, accurate traffic prediction stands as a critical component for managing congestion, planning routes, and enhancing safety. With the advent of Graph Neural Networks (GNNs), researchers have unearthed powerful tools capable of capturing the complex dependencies inherent in traffic systems, which are naturally structured as graphs [1].

GNNs, which extend the principles of deep learning to graph-structured data, have shown promising results in various domains, including traffic management, where they are considered to be state-of-the-art [2]. However, choosing a specific model is heavily contingent upon the quality and quantity of the training data. As no singular model can be best performing on all tasks, considering other criteria such as data availability becomes crucial [3].

This project aims to systematically investigate the effect of data availability and quality by evaluating a GNN model across varying scenarios. Through this analysis, this research aims to identify the base data characteristics that are required by a model to accurately and reliably predict traffic, thereby contributing to more effective traffic management solutions.

This research project explores a fundamental aspect of machine learning: the impact of training data characteristics on model performance. Specifically, it examines how the amount of data available for training and its distance in time to the test data influence the predictive accuracy of GNNs in the context of traffic forecasting.

By bridging the gap between data science and urban planning, this study not only advances our understanding of GNN architectures but also facilitates the development of smarter, data-driven approaches to traffic prediction. In doing so, it addresses a key question:

> What is the minimum amount of data required for a GNN to effectively predict traffic patterns?

The main question this paper aims to answer is:

> What is the effect of reducing the volume of training data on GNNs ability to accurately forecast traffic?

. To answer this questions, multiple sub-questions are asked:

> What is the effect of reducing the number of datapoints available by filtering data out?
>
> How does the distance in time from data in the training set to data in the test set impact the learning process and prediction accuracy of GNNs?

As stated by Jiang et al., "high-quality datasets are expensive to build" [4, p.20] due to the potentially long and expensive process of data gathering. This process entails setting up sensors and maintaining them. Obtaining insights into the effect of reducing the amount of training data has on GNNs predictive performance can potentially lead to a shorter and more cost-efficient data gathering process. This in turn enables more widespread use of GNNs for traffic forecasting tasks in underprivileged areas. Furthermore, changes to the traffic infrastructure need to be recorded and updated within GNNs. This leads to increased expenses during the data gathering process. By shortening this process, the future costs required to update GNNs with new information also diminish.

The approach used to conclude this experiment, the setup and the results are mentioned, followed by ethical considerations of this project. At the end, there is a short discussion of the results, follow by conclusions that can be drawn from this experiment.

2 Related Work

A challenge posed by the data in regard to using GNNs for traffic prediction tasks is called the 'cold-start problem' [4]. As stated by Jiang et al., GNNs usually require a large amount of training data in order to achieve satisfactory predictions. However, the minimum amount of historical data required for GNNs to accurately forecast traffic is rarely discussed, most probably due to the availability of large enough benchmark datasets such as METR-LA, PEMS-BAY and many more [2]. While finding the precise number of data points required per sensor is almost impossible because each model processes data differently, being able to make an educated guess is beneficial when constructing new datasets and when planing to integrate GNNs in real-life scenarios for urban development.

Other studies looked into techniques for reducing the amount of data by removing redundant data from the dataset [5]. Some also delve into the impact of using differently sized time sequences as input and observe the difference in performance metrics [6]. In both studies, it was observed that using more data leads to more accurate results.

More similar studies that looked into the effect of using differently sized datasets found that Long Short Term Memory Networks (LSTMs) experience a dramatic increase in performance when the dataset contains traffic information of at least two weeks, as opposed to using one week [7].

3 Methodology

This section contains a detailed explanation of the experimental approach as well as the setup used to perform the experiments. The first subsection explains the approach chosen to answer the sub-questions posed by this paper. The second subsection delves into the experimental setup used to perform the experiments.

3.1 Approach

This section contains information about how to answer the research questions presented in the section above.

The experiments involve training a GNN model repeatedly with various amounts of training and validation data. To accurately study the effect of the amount of training data, it is important to test how the volume of training data affects the model's accuracy. Also, the distance in time from when the measurements are taken to when the data from the test set is placed, could potentially affect accuracy. Measuring the difference in performance between scenarios that have similar data availability, and with a different time gap between the training set and the test set is also important. This is useful for deploying such models in real world scenarios where resources are sparse, and deployment speed is crucial. Collecting data until a certain expected accuracy is obtained, also implies that the selected model does not need to be retrained or updated to ensure the quality of predictions, avoiding unnecessary data collection and storage until this becomes mandatory due to changes in the local landscape.

Formal Definition

GNNs perform predictions over time series. More precisely, they take a sequence of data arrays that contain traffic information for each sensor at a given point in time and predict what the next sequence of traffic conditions will be using the same format. We will take the whole dataset and create new training sets. These training sets will be used to train the model and compute performance metrics based on the predictions it produces. Formally:

Let $\mathcal{D} = \{\mathbf{x}_i\}_{i=1}^{N}$ be the entire dataset. Define the training set $\mathcal{D}_{train} \subseteq \mathcal{D}$ with size $n = b - a$ such that $\mathcal{D}_{train} = \{\mathbf{x}_i\}_{i=a}^{b}$ and $b \leq 0.8 \cdot N$, where N represents the total number of sensor readings. $\mathbf{x}_i$ represents an array of data that contains the traffic information at index i.

Then, train the GNN model $f(\mathbf{X}; \theta)$ on $\mathcal{D}_{train}$, where θ represents the model parameters and $\mathbf{X}$ is a sequence of consecutive sensor readings.

Using the trained model, the predictions $\hat{\mathbf{Y}} = f(\mathbf{X}_{test}; \theta)$ are obtained on the test set $\mathcal{D}_{test} = \{\mathbf{x}_i\}_{i=N-m}^{N}$, where $m = 0.2 \cdot N$ represents the number of samples in the test set. For a sequence of test inputs $\mathbf{X}_{test}$, the true values $\mathbf{Y}_{test}$ represent the next sequence of sensor readings from $\mathcal{D}$.

The metrics used to measure performance are: mean absolute error (MAE), root mean square deviation (RMSE) and mean absolute percentage error (MAPE).

The performance metrics are calculated based on the predicted values $\hat{\mathbf{Y}}$ and the true values $\mathbf{Y}_{test}$.

$$\text{MAE} = \frac{1}{m} \sum_{i=1}^{m} |\hat{y}_i - y_i|$$

$$\text{RMSE} = \sqrt{\frac{1}{m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2}$$

$$\text{MAPE} = \frac{1}{m} \sum_{i=1}^{m} \left| \frac{\hat{y}_i - y_i}{y_i} \right|$$

Where $\hat{y}_i$ is the predicted value, and y_i is the true value.

Model Selection

The first step in measuring the impact of training data on a GNN model is to select a model. For this purpose, the Decoupled Dynamic Spatial-Temporal Graph Neural Network (D2STGNN) model [8] was selected. This choice was made for multiple reasons. First of all, this model is among the best performing models on multiple publicly available datasets for speed prediction, such as METR-LA and PEMS-BAY [9; 10]. Moreover, it is easy to use and set up.

GNNs use graph structures to model the complex spatial dependencies of traffic forecasting [2]. D2STGNN uses an adjacency matrix to model the graph like structure of the road network, then performs further processing by separating traffic signals into multiple signals. The proposed architecture of the model uses three components. First, there is a decouple block that decomposes the time series of traffic signals into two time series which represent the diffusion and inherent signals. The diffusion block handles the diffusion signals, which represent traffic data spread across the entire network, while the inherent block handles the inherent signals that represent inherent traffic patterns at specific locations. Finally, a fusion model is used to integrate the results from the diffusion and inherent blocks and make final traffic predictions.

Dataset Selection

The second step is selecting the dataset. That dataset is METR-LA because it is among the most popular datasets for training [2; 4]. The dataset contains traffic information measured at 5 minutes intervals from 207 loop detectors on the highways of LA County, spanning over 4 months from the 1st of March until the 30th of June in the year 2012 [8; 11].

3.2 Experimental Setup

This subsection presents the data processing employed by this experiment, as well as the specifics of executing the code. The first part details how the dataset was chosen, how it was processed and how new datasets were created. The second part details how to train the selected model and how this process was performed in the scope of the experiment.

Data Processing

This part explains how the initial the dataset has been processed so that new sets are created.

First of all, the last 20% of the data from METR-LA is always used as the test set. This is the percentage of test data

used to measure the performance when training the model over METR-LA used by its authors [8] so that the accurate reproduction of the model can be checked.

From the rest of the data, new sets were created so that they cover measurements spanning over two months, one month, half a month and finally one week intervals from March and April of the year 2012. These time frames are rough estimations of the lengths used. Because four weeks contain a total of 28 days, some of the datasets that represent a week contain 8 days instead of 7. This has been done so that the datasets of roughly the same size cover March and April. For the exact time intervals used for each dataset, including the MAE, RMSE and MAPE, see Appendix A.

The filtering process is performed by reading the initial dataset and indexing the data on date and time. Then, a copy of the dataset is filtered based on a date range and saved as a new dataset. Both the initial dataset and the generated filtered datasets are saved using the Hierarchical Data Formats (HDF) used to store large amount of data, more specifically using H5 file format.

The model is then trained with access to the filtered datasets, in the form of training and validation sets, and finally tested against the test set. From each of the filtered datasets, 90% of the data is used for generating the training set, while the remaining 10% is used to generate the validation set. Both the training and validation sets are generated by inputting the filtered datasets into the 'generate_training_data' script provided inside the D2STGNN Python project publicly available[1]. The script reads the dataset and generates sequences of input and output data of fixed lengths. These generated sequences are then split into training, validation and test sets. The script used in this experiment to generate the training and validation sets is a copy of the original script, where the number of test samples is set to 0 and the number of training samples is set to 0.9 of the total number of samples. The number of validation samples is computed as the number of total samples, from which we subtract the number of training and test samples.

Code Execution

It is also important to note the specifics of training a GNN model. In this case, all the models trained used the same computer for training and the hyper parameters are identical for all the models. D2STGNN uses Adam as the optimiser, with the initial learning rate set to 0.001 over input and output sequences of length 12 [8]. For a more detailed overview of parametrisation, refer to either the paper or to the open repository of the author. The exact configuration files used for the experiments can be found on the publicly available GitHub repository[2].

The execution of the code was performed on the Delft-Blue Supercomputer of Delft High Performance Computing Center (DHPC) [12]. The code execution is performed on GPU nodes with four NVIDIA A100 GPUs with 80 GB video RAM each. For the specific scripts used to execute the code on the DelftBlue Supercomputer refer to the last mentioned repository.

[1]https://github.com/zezhishao/D2STGNN

[2]https://github.com/AlexPacurar01/Research-Project-Code

4 Results

First, it is important to compare performance metrics from datasets that use similar time frames. As seen in both Figure 1 and Figure 2, the models trained on datasets spanning over similar time frames have close performance metrics. The exceptions are the dataset from the fourth week of March and the dataset over the first half of April. In Figure 1 the MAE of the predictions are mostly similar, while there are bigger differences in RMSE and MAPE being recorded. The best results are recorded by the model that used the third week of April (from the 16th until the 23rd), while most models trained over one week time frames from April outperform those trained using data from March, both models trained using the first and second week from March outperform the model trained with data from the last week of April on all metrics. In Figure 2 the model that is that trained with data from the last half of April outperforms those trained with data from March by a very small, while both models that used data from the month of March perform noticeably better than the model trained with data from the first half of April.

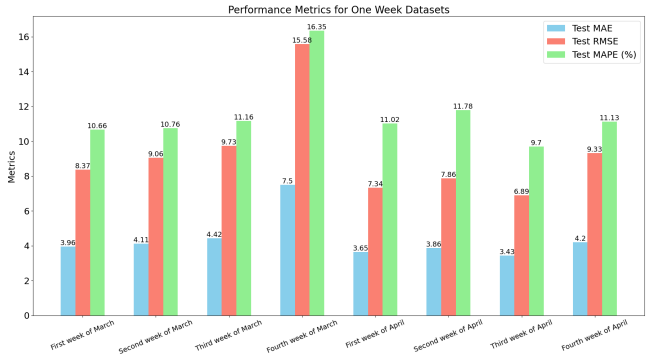

Figure 1: Performance metrics of training the model with data from different one week time frames.

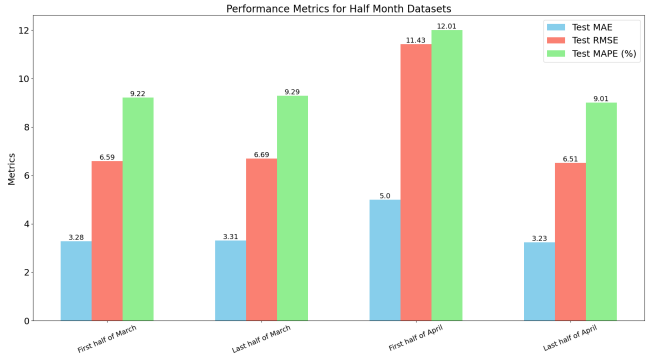

Figure 2: Performance metrics of training the model with data from different half month time frames.

When excluding the two scenarios that perform worse than expected by a larger margin, the differences in accuracy are not significant and most probably occur due to factors such

as data sparsity and how close does the data from the training set resemble the data from the test set.

On the other hand, these two exceptions are most likely due to errors in sensor readings. In the initial dataset, there are large blocks of zero values that most likely affect the model's ability to correctly predict future traffic. While in general, values of zero occur seemingly at random, in some cases the sensors do not perform any measurements for long periods of time. Models that are trained with data that contains such cases tend to perform noticeably worse than models trained on datasets that do not contain continuous blocks of zero values.

When comparing the performance metrics of training the model with data from March and with data from April, it is clear that the model trained only with data from March outperforms, as seen in Figure 3. This is also most probably related to the large blocks of missing values. The measurements from the first half of April contain more of these problematic readings, and for longer periods of time, than the measurements from March. However, when the model is trained with more data, the difference in performance is smaller because the blocks of missing values represent a smaller percentage of the training data.

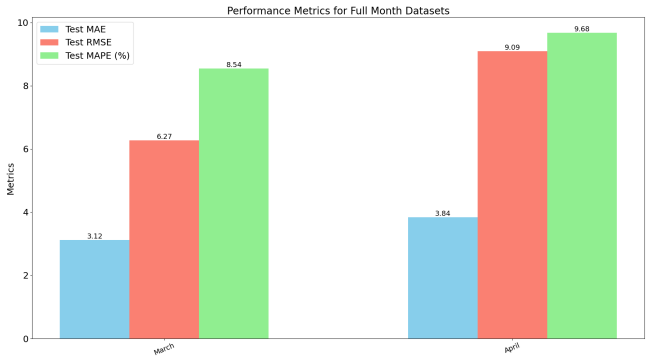

Figure 3: Performance metrics of training the model with data from March and April respectively.

Considering the results from above, it is safe to conclude that the distance in time from data used in the training set to the data used in the test set is not proportional to the performance of the model. The amount of data, as well as the quality of the measurements, are more important in determining the impact training data has on a GNNs ability to accurately predict traffic.

The average impact of changing the size of the time frame used to filter data is more apparent in Figure 4. Using more data leads to better accuracy and to a lower standard deviation. With an increase in training data, continuos sensor maintenance becomes less important because erroneous or missing values represent a smaller percentage of the data used for training. The model trained with all the data recorded in March and April outperforms all models trained with less data over all metrics. Moreover, the model trained using training data from both months outperforms the models trained with data from one-week intervals by more than one standard deviation for all metrics and has a MAPE value lower than the average of any group by more than the standard deviation.

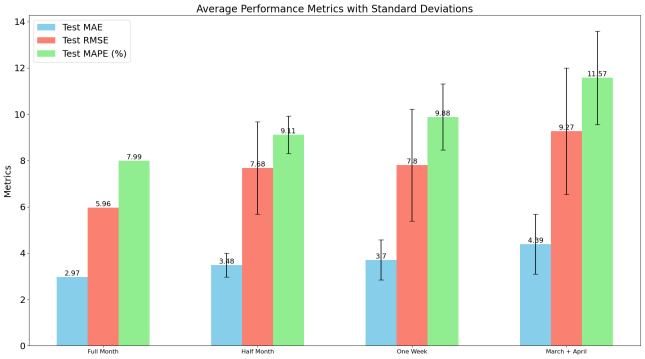

Figure 4: Average performance metrics of training the model with data from similarly sized time frames of one week, half a month and one month respectively. Standard deviation is indicated by the error bar.

More in depth data that contains the exact performance metrics recorded by training the model with data from each dataset and the time frame used to filter the datasets can be found in Appendix A.

5 Responsible Research

In conducting this research on GNNs for Traffic Forecasting, several considerations are made to ensure the study is conducted responsibly and ethically.

5.1 Ethical Considerations

There are two main considerations for this research: training GNN models requires resources and ensuring data privacy and its correct usage.

Training GNN models, especially using a supercomputer, requires significant computational power. This raises concerns about energy consumption and environmental impact. To address this, I strived to efficiently use available resources to minimize unnecessary computational waste.

The dataset used in this research is METR-LA, which is a publicly available traffic information dataset. It does not contain personal information or directly involve human subjects, thus mitigating privacy concerns. However, responsible usage involves acknowledging the sources of this dataset and ensuring it is used strictly for the purposes of this research.

5.2 Research Integrity

This research adheres to the guidelines and standards set by TU Delft, ensuring that all practices align with the accepted norms for ethical and responsible AI research. This includes transparency in reporting methodologies, results, and potential biases. All research activities are conducted in accordance with the TU Delft Code of Ethics[3] and the Netherlands

[3]https://www.tudelft.nl/en/student/legal-position/education-regulations/code-of-ethics

Code of Conduct for Research Integrity[4].

To ensure the integrity and reproducibility of the research findings, I strived to make the experiment easy to reproduce by recording decisions taken and making the code publicly available. This includes the selection of datasets, the configuration of the GNN models, the training processes, and the evaluation metrics. By providing detailed explanations of the methods and rationales behind each decision, I aim to maintain transparency and allow others to replicate the study if desired.

The accuracy and validity of the research findings are critical. This involves conducting thorough testing and validation of the GNN models on multiple datasets with varying sizes. Results are compared and analysed to determine the minimum data requirements and the impact of data measurement on prediction accuracy. Any anomalies or unexpected findings are investigated and reported.

6 Discussion

Considering the results from the previous sections, it is interesting to observe how while on average using a larger time frame of data leads to better results, this is not true for every particular case. Most notably, training the model over the first half of April leads to worse results than training on only one week in most cases. This is clearly visible when comparing Figure 1 and Figure 2.

When looking at all the data from METR-LA, some of the sensor values are missing. This is apparent when looking at the 26th of March or at the 8th of April for example. All the sensors recorded an average speed of 0 miles per hour for long periods of time, which is very unlikely to happen for all sensors at once. This indicates that most probably there were issues with the data measurement process. As an effect, some datasets have an increased percentage of zero values (sparsity) compared to others. It is known that GNNs perform better when trained with less sparse data and can profit from using missing data imputation techniques. As stated by Gadelho: "GNNs can benefit from these techniques, especially when used with VCI and METR-LA datasets" [13, p.75]

While the data sparsity percentages are similar across most of the datasets, both models that performed considerably worse than expected for their respective category had a more sparse training dataset. The dataset containing information from the fourth week of March has around 23.5% sparsity, while the second highest sparsity percentage recorded is approximately 11.8% recorded in the dataset from the first week of April. The average sparsity for one week datasets, excluding the fourth week of March, is approximately 7%. The sparsity of all the data excluding the data used in the test set is very close to 8.11%.

However, the percentage of zero values is not the only factor relevant to sparsity. When comparing the performance of the model when trained using time frames of one month in Figure 3 it is clear that using the data from the month of March leads to better results. The sparsity of data from the month of March is almost exactly 9%, while that of the data from the month of April is approximately 8.38%. The most probable cause for this is the distribution of missing values within the datasets. While few sensors not working is inconvenient, sometimes the dataset contains some large time frames where all sensors record values of zero. On the 26th of March no sensor registered any cars passing for seven hours and 40 minutes. On the 8th and 9th of April, the sensors recorded only values of zero for a total of 29 hours and 10 minutes. This is not apparent in the performance metrics of training the model over one week in April because the first week uses data until the 8th, so this large block of missing values is split over two different datasets. This would also explain why training over the first half of April leads to worse results than training over a week on average.

[4]https://www.nwo.nl/en/netherlands-code-conduct-research-integrity

7 Future Work

Due to the limitations of this research project, such as the use of a singular initial dataset, a singular GNN model and lack of data imputation techniques, there are multiple aspects of this experiment that could be changed so that new information is derived.

First of all, this experiment can be reproduced with any model, so observing how different GNNs behave when trained with less and less data could help obtain more insight into how GNNs behave for the task of traffic forecasting.

Moreover, using different initial datasets is also beneficial. All the models are trained with data from the same location. Using multiple datasets that contain traffic information from various areas and with different sparsity percentages is very useful for generalising this experiment.

In addition, observing how GNNs behave on smaller and smaller datasets when paired up with imputation techniques would most probably give results closer to what can potentially be achieved in real life scenarios.

8 Conclusion

Considering all that has been said thus far, using training sets that span over shorter time frames does not lead to a significant decrease in performance on average. The expected loss of accuracy is of less than 2 miles per hour when using a training dataset that contains sensor measurements from one week instead of two months. This is true in cases where sensor errors do not lead to a very sparse dataset.

With this information in mind, creating new datasets and using GNNs in urban areas that did not previously have access to this technology should be faster to implement since the time required for gathering data after setting up sensors can be reduced without drastically decreasing the accuracy of the predicted traffic. However, ensuring the quality of the dataset becomes increasingly more important as the time spend to record training data decreases. In cases where maintenance costs are high, it is better to focus on properly maintaining the sensors for a shorter period of time so that the training set is of high quality and then perform occasional maintenance checks than to keep up with the high cost of constant maintenance for longer periods of time in exchange for a small performance increase.

References

[1] I. R. Ward, J. Joyner, C. Lickfold, Y. Guo, and M. Bennamoun, "A practical tutorial on graph neural networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–35, 2022.

[2] W. Jiang and J. Luo, "Graph neural network for traffic forecasting: A survey," *Expert Systems with Applications*, 2022.

[3] K. Lee *et al.*, "Short-term traffic prediction with deep neural networks: A survey," *IEEE Access*, 2021.

[4] W. Jiang, J. Luo, M. He, and W. Gu, "Graph neural network for traffic forecasting: The research progress," *ISPRS International Journal of Geo-Information*, vol. 12, no. 3, 2023. [Online]. Available: https://www.mdpi.com/2220-9964/12/3/100

[5] Z. Gao, X. Yang, J. Zhang, H. Lu, R. Xu, and W. Diao, "Redundancy-reducing and holiday speed prediction based on highway traffic speed data," *IEEE Access*, vol. 7, pp. 31 535–31 546, 2019.

[6] Z. Song, Y. Guo, Y. Wu, and J. Ma, "Short-term traffic speed prediction under different data collection time intervals using a sarima-sdgm hybrid prediction model," *PloS one*, vol. 14, no. 6, p. e0218626, 2019.

[7] E. Doğan, "Lstm training set analysis and clustering model development for short-term traffic flow prediction," *Neural Computing and Applications*, vol. 33, no. 17, pp. 11 175–11 188, 2021.

[8] Z. Shao, Z. Zhang, W. Wei, F. Wang, Y. Xu, X. Cao, and C. S. Jensen, "Decoupled dynamic spatial-temporal graph neural network for traffic forecasting," *arXiv preprint arXiv:2206.09112*, 2022.

[9] "Papers with Code - METR-LA Benchmark (Traffic Prediction)." [Online]. Available: https://paperswithcode.com/sota/traffic-prediction-on-metr-la?p=spatio-temporal-graph-convolutional-networks

[10] "Papers with Code - PEMS-BAY Benchmark (Traffic Prediction)." [Online]. Available: https://paperswithcode.com/sota/traffic-prediction-on-pems-bay?p=190600121

[11] Y. Li *et al.*, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *International Conference on Learning Representations*, 2018.

[12] Delft High Performance Computing Centre (DHPC), "DelftBlue Supercomputer (Phase 2)," https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2, 2024.

[13] A. C. M. Gadelho, "Application of graph neural networks in road traffic forecasting for intelligent transportation systems," 2023.

A Appendix A

Time Frame	Start Date	End Date	Test MAE	Test RMSE	Test MAPE
METR-LA	2012-03-01	2012-06-04	2.88	5.80	7.78
March	2012-03-01	2012-03-31	3.12	6.27	8.54
April	2012-04-01	2012-04-30	3.84	9.09	9.68
March + April	2012-03-01	2012-04-30	2.97	5.96	7.99
First half of March	2012-03-01	2012-03-15	3.28	6.59	9.22
Last half of March	2012-03-16	2012-03-31	3.31	6.69	9.29
First half of April	2012-04-01	2012-04-15	5.00	11.43	12.01
Last half of April	2012-04-16	2012-04-30	3.23	6.51	9.01
First week of March	2012-03-01	2012-03-07	3.96	8.37	10.66
Second week of March	2012-03-08	2012-03-14	4.11	9.06	10.76
Third week of March	2012-03-15	2012-03-21	4.42	9.73	11.16
Fourth week of March	2012-03-22	2012-03-28	7.50	15.58	16.35
First week of April	2012-04-01	2012-04-08	3.65	7.34	11.02
Second week of April	2012-04-09	2012-04-15	3.86	7.86	11.78
Third week of April	2012-04-16	2012-04-23	3.43	6.89	9.70
Fourth week of April	2012-04-24	2012-04-30	4.20	9.33	11.13

Table 1: Performance metrics for various time frames

Note that in the above table, the exact last date used by the training and validation set is 2012-06-04 04:45:00. The first time stamp used in the test set is 2012-06-04 04:50:00.