Effectiveness of Graph Neural Networks and Simpler Network Models in Various Traffic Scenarios

Graph Neural Networks for Traffic Forecasting

Wiktor Grzybko
Supervisor: Elena Congeduti

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: Wiktor Grzybko
Final project course: CSE3000 Research Project
Thesis committee: Elena Congeduti, Lilika Markatou

Abstract

Traffic forecasting is key to improving urban transport and reducing congestion and pollution. While advanced models like Graph Neural Networks (GNNs), can capture complex patterns in traffic flow, they are resource-intensive and do not scale well. This problem can be mitigated by using simpler models that are less influenced by the size of the road network, making them more practical for real-world applications. This study investigates whether simpler network-based models, particularly Long Short-Term Memory (LSTM) networks, can match or surpass the performance of GNNs, such as the Diffusion Convolutional Recurrent neural network (DCRNN), in specific scenarios. Using popular benchmark datasets, we compared the performance of the LSTM and DCRNN models under different conditions, including different sensor distributions and prediction horizons. The results indicate that while DCRNN highly outperforms LSTM with numerous sensors and longer prediction horizons, LSTM gives promising results with fewer sensors and shorter horizons. In this scenario, the difference in performance is minimal regardless of the location of sensors, also offering significant computational efficiency. These findings suggest that LSTM models may be a practical alternative for traffic forecasting in resource-constrained scenarios, providing a path to more efficient urban traffic management.

1 Introduction

Traffic forecasting is essential for improving urban transport systems. Accurate predictions can optimise traffic signal timings, provide real-time route recommendations and improve public transport schedules. However, traffic forecasting is difficult due to the complexity of the data and unpredictable events, such as accidents and weather changes, that can drastically affect traffic flow. Historically, simpler models like Autoregressive Integrated Moving Average (ARIMA) [15] or Recurrent Neural Networks (RNNs) [14] have been used for traffic forecasting, but often have difficulty learning the relationships between different parts of the road network. Recently, more advanced methods like Graph Neural Networks (GNNs) have been introduced [4]. Unlike other models, GNNs use the structure of road networks, which can be represented as graphs. In these graphs, the sensors installed on the roads (such as loop detectors) act as nodes and the roads themselves serve as edges. By taking these graphs as input, GNNs learn the interactions between the different nodes within the network [4]. This feature makes them more efficient in capturing traffic flow patterns.

One of the first GNN models to achieve outstanding results in predicting traffic volumes in large, complex metropolitan areas was the Diffusion Convolutional Recurrent Neural Network (DCRNN) [6]. The DCRNN model integrates a diffusion convolutional layer with a recurrent neural network and takes the mentioned graphs as input. This combination enables the processing of complex relationships in data, such as traffic flows, through the use of graph-based spatial information and time series analysis, making it highly effective in dynamic forecasting tasks [6].

However, GNNs are complex and costly to train. Previous researches highlight GNNs' inefficiencies in large-scale or densely connected graphs, posing scalability challenges for real-time applications due to their quadratic computational complexity [7]. This opens up the field for research into simpler models that perform even better [7, 11] and raises questions about their necessity in all traffic forecasting scenarios, especially when simpler models can be effective in less demanding conditions.

Models based on a simpler neural network structure have been successfully applied to similar prediction tasks for a long time and require much less computational training time. There are several possibilities for such models discussed in many publications, some of which are Long Short-Term Memory (LSTM), Feedforward Neural Network (FNN) and RNN [13]. Based on the analysis of traffic prediction models, LSTM networks [3] stand out as a better choice compared to traditional RNNs and FNNs for several reasons. While simple RNNs are designed to handle sequential data and can learn temporal dependencies, they suffer from problems such as fading gradients that limit their ability to effectively capture long-term dependencies. LSTMs deal with this problem by incorporating memory cells and gating mechanisms that allow them to maintain and update information for longer periods, thereby better capturing both short-term and long-term dependencies [14]. FNNs, on the other hand, do not consider temporal sequences, treating each input independently, which is a significant limitation in traffic forecasting where temporal dynamics are crucial. This makes LSTM the preferred model for traffic forecasting tasks, offering significant improvements in prediction accuracy and robustness compared to traditional RNNs and FNNs [13].

A very comprehensive comparison of different forecasting models in various traffic scenarios has been conducted in [8, 9, 16]. These scenarios may include differences in the number of sensors available, the type of area in which the sensors are placed (e.g. highways or urban areas) and the prediction horizon (e.g. 5 or 15 minutes in the future). In the article [8], the authors even tested the suitability of deep learning models for traffic prediction, trying to find scenarios where simpler models are a better choice. However, despite including many deep learning models in the experiments, the consideration of GNNs is lacking. Those that consider GNNs [6, 17] tend to focus on entire datasets rather than specific scenarios. Other studies have examined the predictive capabilities of the DCRNN by testing it with varying sensor locations. These studies aimed to understand how different road infrastructure scenarios and the availability of sensors impact the model's performance. Based on the received results, the author suggested that the model is very sensitive to sensor locations. Selecting sensors from different parts of the network, even if they are selected non-randomly and locally, can lead to significant differences in results [10].

Extensive studies comparing GNNs to simpler network models in specific traffic scenarios are lacking. This gap highlights the need to investigate whether simpler models can provide comparable performance to GNNs in certain traffic scenarios. Therefore, the main contribution of this paper is to explore scenarios where simpler models, in particular LSTM networks, could replace more complex GNNs such as DCRNN in the context of traffic forecasting.

Consequently, the research question of this paper is: **Are there possible scenarios in which models based on simpler network structures are more effective than GNNs in traffic forecasting?** To address this question, the following hypothesis was formulated: *LSTM models should achieve comparable performance to GNN models in traffic scenarios where spatial interconnections are minimal, such as highways with few intersections, and the number of available sensors is limited.*

The paper is structured as follows: Section 2 describes the models used in the experiments and the datasets, formally introduces the problem and presents the traffic scenarios used in the evaluation. Section 3 describes the experiments performed in detail and provides an overview of the results obtained. In Section 4, the discussion focuses on the practices of Responsible Research. The final conclusions are presented in Section 5. The paper ends with Section 6, which suggests potential future research directions.

2 Methodology

This section focuses on the description of the LSTM and DCRNN models and the research methodology. A deeper look at the architecture of these models is presented, as well as a description of the problem, data sets and evaluation metrics.

2.1 Models Description

Long Short-Term Memory

LSTM networks are a type of RNN but have been designed to overcome the limitations of traditional RNNs. In particular, they address the problem of vanishing gradient [2], which hinders the learning of long-term dependencies. LSTMs use a gating mechanism to regulate information flow, maintaining and updating the memory cell state over long sequences, making them effective for time series forecasting [14].

LSTM architecture contains three primary gates: the input gate, the forget gate, and the output gate. These gates control the information added to or removed from the cell state, retaining important data over extended periods while discarding irrelevant information. This mechanism enables LSTMs to capture both short-term and long-term dependencies in long sequential data.

To obtain multi-step predictions we applied a special mechanism to LSTM forward pass. Firstly, the LSTM processes the input sequence and updates the hidden and cell states. The output is then processed through a fully connected layer to generate the next time step prediction. This prediction is appended to the output sequence and replaces the oldest value in the input sequence, effectively shifting the input sequence window forward. This process is repeated for a specified number of steps (horizons), allowing the model to make sequential forecasts by iteratively using its previous predictions as part of the new input. An example of this forecasting procedure can be seen in Figure 1, where X_t and $\hat{X}_t$ represent the actual and predicted values at time step t, respectively.

An in-depth understanding of the architecture of the LSTM has been presented in previous studies [14] [3], providing a solid foundation for the present research.

$$
\begin{aligned}
\left[X_{t-2}, X_{t-1}, X_t\right] &\quad\rightarrow\quad \left[\hat{X}_{t+1}\right] \\
\left[X_{t-1}, X_t, \hat{X}_{t+1}\right] &\quad\rightarrow\quad \left[\hat{X}_{t+2}\right] \\
\left[X_t, \hat{X}_{t+1}, \hat{X}_{t+2}\right] &\quad\rightarrow\quad \left[\hat{X}_{t+3}\right]
\end{aligned}
$$

Figure 1: An example of the forecasting procedure. At each step, we update the input sequence by removing the oldest value and adding the latest prediction as the most recent value. In this example, both the input sequence length and the prediction horizon are set to 3.

Diffusion Convolutional Recurrent Neural Network

DCRNN is an advanced GNN model designed to address the challenges of spatio-temporal traffic forecasting by combining the strengths of convolutional neural networks and RNNs. DCRNN leverages diffusion convolution to capture spatial dependencies and RNNs to model temporal dependencies. The key difference between DCRNN and LSTM is that DCRNN takes graphs as input and explicitly models traffic as a diffusion process over a road graph, reflecting the actual structure and dynamics of the road network [6].

DCRNN's architecture consists of two main components: the diffusion convolution layer and the recurrent layer. The diffusion convolution operation captures the spatial relationships between traffic sensors, modelling traffic flow as a diffusion process on a directed graph. This diffusion process is characterised by random walk on the graph and state transition matrix [6]. To model temporal dependencies, the authors proposed a Diffusion Convolutional Gated Recurrent Unit (DCGRU), a version of the standard Gated Recurrent Unit (GRU) [1], which is another variant of RNNs. The main difference between DCGRU and GRU is the replacement of matrix multiplications in the GRU with diffusion convolution operations [6]. This combination enables DCRNN to provide highly accurate traffic forecasts.

Similarly to the LSTM model, a detailed explanation of the architecture of the DCRNN components is not included in this study, as they are described in depth in the original paper [6].

2.2 Datasets

This research uses the METR-LA and PeMS-BAY datasets, which are widely used benchmarks in traffic forecasting studies. The METR-LA dataset (Figure 2) contains traffic speed data collected from 207 loop detectors on freeways in Los Angeles County, with data from 1 March 2012 to 30 June 2012. The PeMS-BAY dataset (Figure 3), on the other hand, consists of traffic information from the San Francisco Bay Area collected with 325 sensors from 1 January 2017 to 31 May 2017. Both datasets aggregate traffic speeds in miles per hour in 5-minute windows. These datasets are appropriate for this study as they were originally used to demonstrate the superior performance of DCRNN compared to simpler models. Thus similarly to those experiments performed in [6] we split the data so that 70% of the data is used for training, 20% is used for testing and the remaining 10% is used for validation. Additionally, the data was Z-score normalized.

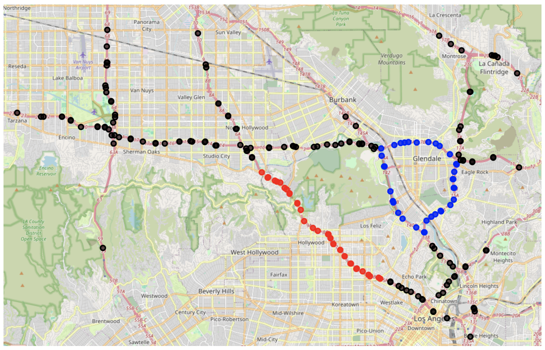

Figure 2: Map showing all sensors from METR-LA dataset with two different study areas selected. Sensors located along the same road are represented by red dots. The area represented by blue dots represents sensors placed on different roads with common intersections.

2.3 Problem Description

The objective of this study is to compare DCRNN and LSTM in different scenarios by predicting future traffic speeds over different periods using historical traffic data. Formally, given a sequence of historical traffic observations $\{X_{t-n+1}, \ldots, X_{t-1}, X_t\}$, the goal is to predict the future traffic speeds $\{X_{t+1}, X_{t+2}, \ldots, X_{t+h}\}$, where n is the input sequence length, h is the prediction horizon and t is a time step. This study will explore the differences in performance of LSTM and DCRNN models under different conditions, such as varying the number and locations of sensors as well as different numbers of horizons to be predicted.

Experiment Scenarios

Experiments were conducted to assess how well the models predict traffic speeds in the following scenarios:

- Different prediction horizons: 5, 10, and 15 minutes ahead
- Different numbers of sensors: 35, 20, 10, 5, and 1 sensor.
- Different road configurations:
  - Sensors from the same road
  - Sensors placed on different roads with shared crossroads

All of the above scenarios were tested in both datasets to obtain more reliable results. Figure 2 shows all sensors from the METR_LA data set with two distinguished areas of roads. The red area represents a subset of sensors from the same highway, while sensors from 3 different roads are blue. For each of these areas, 5 different subsets of sensors were selected as described above, and then both DCRNN and LSTM were trained to predict the 3 horizon steps. The same approach was applied to the PeMS-BAY dataset (Figure 3).

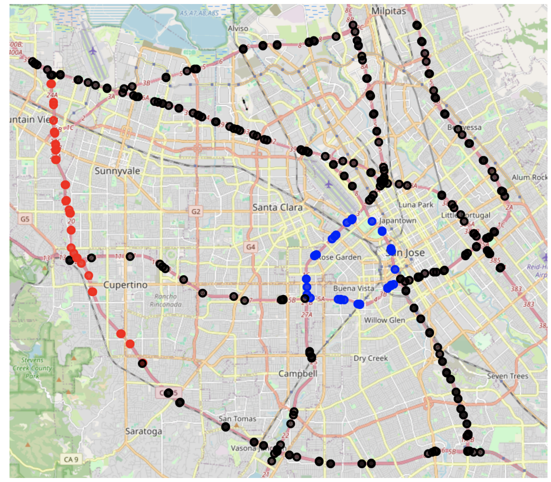

Figure 3: Map showing all sensors from the PeMS-BAY dataset with two different study areas selected. Sensors located along the same road are represented by red dots. The area represented by blue dots represents sensors placed on different roads with common intersections.

2.4 Performance Metrics

The performance of the LSTM and DCRNN models will be evaluated using Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics are defined as follows:

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(\hat{X}_t - X_t)^2} \tag{1}$$

$$\text{MAE} = \frac{1}{N}\sum_{t=1}^{N}|\hat{X}_t - X_t| \tag{2}$$

$$\text{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{\hat{X}_t - X_t}{X_t}\right| \tag{3}$$

where:
- N: The total number of observations.
- $\hat{X}_t$: The predicted value of the observation at time step t.
- X_t: The actual value of the observation at time step t.

These metrics are useful for assessing the accuracy of traffic forecasting models, as they provide insight into the model's prediction errors. RMSE highlights larger values and errors, making it more sensitive to outliers. Formula 2 shows the MAE metric which provides a simple average error figure in the same unit as the data. MAPE (Formula 3) is useful for understanding the relative error in percentage terms, which is particularly important for traffic data where the speed scale can vary widely. However, it has some limitations. One is the vulnerability to values close to zero of the actual data becoming very large or even undefined. To avoid this, zero values and nulls were left off during the calculation of these metrics for both models to retain only the most valuable information. While all the above metrics were used to assess the models, MAE was used additionally for the training as a loss function.

In addition to these metrics, the study will analyze the performance improvement over multiple training epochs and average epoch training times.

3 Experimental Setup and Results

This part of the paper explains the details of the experimental setup made for the experiments, how the models were tested, and the results obtained from the experiments.

3.1 Experimental Setup

The experiments were done on a MacBook Pro 2017 with 16GB of RAM, a 2.6 GHz 6-Core Intel Core i7 processor, and macOS 13.6.3. For the development and execution of the models, Python 3.9 was used, and the models were implemented using the PyTorch[1] library.

Model Configurations

DCRNN: The model is a Pytorch reproduction of the original DCRNN described in the paper "Diffusion Convolutional Recurrent Neural Network: Data-Driven Traffic Forecasting" [6]. The code of the proposed implementation can be found in the public GitHub repository[2]. For the experiments, the sequence length and horizon were set to 3, with patience of 20 epochs for early stopping. Patience for early stopping refers to the number of epochs with no improvement in validation loss before training is halted to prevent overfitting.

The **LSTM** was constructed using 256 LSTM units, 2 recurrent layers, and a dropout rate of 0.5 during training. This model was trained for 500 epochs with a patience of 10 epochs using the Adam optimizer, with the learning rate set at 0.001. Furthermore, after initial experiments, the number of LSTM units in the model was reduced to 50 for subsets with 5 and 1 sensors. This adjustment led to better results for smaller subsets and accelerated the learning process. The implementation of the LSTM is also publicly available on GitHub[3].

It should be noted, that the main purpose of this experiment is to compare the capabilities of GNNs and simpler neural network model in traffic forecasting, not to optimise their implementation. We chose the DCRNN and LSTM models because of their popularity in this research area and as reliable representations of the architectures of these models.

The subsets of sensors used in the experiments can be found in the Google Drive[4].

3.2 Methods of performance evaluation

As mentioned before, the goal of the experiments was to compare the results of DCRNN and LSTM in different scenarios and find the ones when the LSTM performed the best relative to DCRNN. Therefore, the objective is not to find scenarios in which the LSTM performs best but to find situations in which the difference (see Formula 4) between the results of DCRNN and LSTM is minimal or even when the LSTM's performance is better than that of the DCRNN. The starting point for the experiment was the hypothesis which states that LSTM should achieve the closest results to the DCRNN in a situation where the number of sensors on the road is limited and their distribution alone is restricted to a single road. In total, both models were trained for 20 different subsets of sensors (for each of the 2 datasets, we selected 2 separate areas for which we chose 5 subsets of sensors).

$$Error_{Difference} = Error_{LSTM} - Error_{DCRNN} \quad (4)$$

MAE			
Scenario	DCRNN	LSTM	Difference
METR-LA	2.77	4.8918	2.1218
PEMS-BAY	1.38	2.2716	0.8916
RMSE			
Scenario	DCRNN	LSTM	Difference
METR-LA	5.38	12.25	6.87
PEMS-BAY	2.95	4.76	1.81
MAPE			
Scenario	DCRNN	LSTM	Difference
METR-LA	7.30%	9.81%	2.51%
PEMS-BAY	2.90%	5.16%	2.26%

Table 1: All sensors comparison of DCRNN and LSTM for 15 minutes prediction. The results for DCRNN were reported in [6]. The results for LSTM come from performed experiments.

3.3 Results

We divide our observations into 3 parts based on the particular scenario and training time analysis. As a good reference point to our results, in Table 1 we provide the performance of both models for the full datasets. The performance results for DCRNN visible in the table were taken from its original paper [6] and results of LSTM come from our experiments. In the table, we can see the superiority of DCRNN over LSTM in each metric. Particularly important is the "Difference" column showing the differences between the results of the two models computed according to Formula 4 and it will be the main point of focus in further research. This difference is particularly noticeable for the METR-LA dataset, which is generally considered to be more complicated in the context of traffic forecasting [6].

Multi-Step Prediction Performance

The first scenario discussed in this section is the different forecast horizons. To obtain the graphs, the average performance differences over 5 subsets of sensors for each road scenario and metric were calculated as follows: for a specific horizon, the difference error from each subset was summed and then divided by the number of subsets. An example for Horizon 1 can be found in Formula 5, where Avg_{H1} is an average difference error for Horizon 1 and $D_{Si,H1}$ is a difference error for a subset i and horizon 1.

Figures 4 and 5 show the differences in LSTM and DCRNN results for 3 different prediction horizons (5, 10 and 15 minutes) for sensors deployed on the same and different roads (red and blue areas in Figures 2 and 3), respectively. In both figures, we see that the DCRNN systematically increases its lead in each metric over the LSTM model with an

[1] https://pytorch.org

[2] https://github.com/chnsh/DCRNN_PyTorch.git

[3] https://github.com/wiktorr22/research_project_LSTM.git

[4] sensor subsets

increasing horizon independent of the dataset or area type. However, this increase is barely visible for MAE and RMSE metrics for METR-LA dataset in Figure 5. Moreover, in this part of the experiment, both models performed the worst on average (see Table 2). This observation may indicate that the GNN model does not make effective use of its ability to learn spatial dependencies and perform similarly to the LSTM under challenging road conditions. This will be tested in more detail in the next section.

The results show that the performance gap increases as the forecast horizon increases. This indicates the need to examine other scenarios based on the shortest 5-minute forecast, as the difference between the two models is minimal.

$$Avg_{H1} = \frac{D_{S1,H1} + D_{S2,H1} + D_{S3,H1} + D_{S4,H1} + D_{S5,H1}}{5} \tag{5}$$

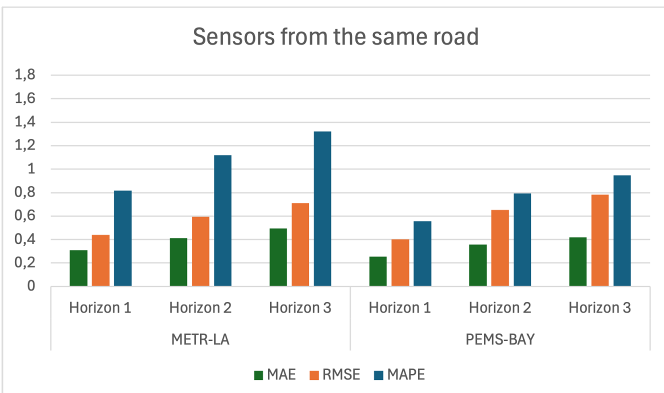

Figure 4: Difference in models' performance in different prediction horizons on the same roads (red areas in Figures 2 and 3).

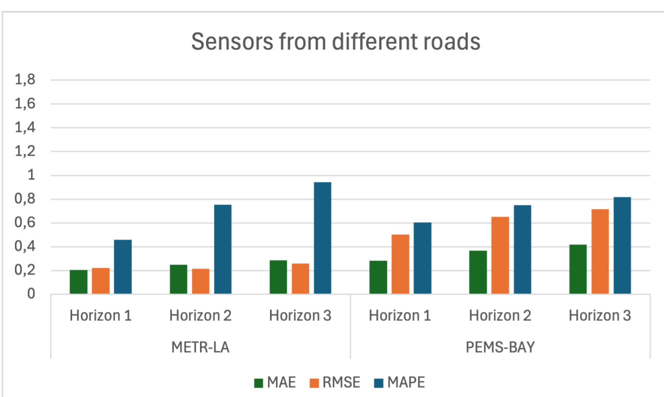

Figure 5: Difference in models' performance in different prediction horizons on different roads (blue areas in Figures 2 and 3).

Analysis by Number of Sensors

In this scenario, we investigate the impact of sensor availability in different network areas (Figures 2 and 3). Furthermore, as the differences in the results for horizon 1 are minimal, this analysis will focus only on the results obtained for this specific horizon.

	MAE		RMSE	
	DCRNN	LSTM	DCRNN	LSTM
METR-LA				
Same Road	2.3487	2.7538	0.0551	0.0659
Diff Roads	3.0300	3.2760	0.0739	0.0810
PeMS-BAY				
Same Road	1.1367	1.4807	0.0222	0.0298
Diff Roads	1.2082	1.5814	0.0165	0.0222

Table 2: Average performance results of DCRNN and LSTM models for different road scenarios. (The entire table can be found in the appendix A as Table 4)

Figure 6 displays the performance metric differences between LSTM and DCRNN for varying number of sensors (red areas in Figures 2 and 3). We can immediately see the relationship of these differences with the number of sensors used. A subset of just 35 sensors significantly offsets the differences observed for entire datasets (see Table 1). A noticeable, constant decrease in the difference occurs up to 10 sensors for both datasets, where the difference between the two models is already very small compared to the differences for the whole datasets and equals 0.234 MAE, 0.23 RMSE and 0.74% MAPE for METR-LA and 0.1868 MAE, 0.26 RMSE and 0.42% MAPE for PeMS-BAY. For subsets of 5 and just 1 sensor, some interdependence is still observed, but it is not as consistent. This might be because the sensors do not influence each other as significantly in the DCRNN model when only 5 sensors are available. These sensors may be from the same road but located in different directions, resulting in a larger road distance (the distance along existing roads rather than simple Euclidean distance). Consequently, the prediction is mainly made by the recurrent component of the model, which is very similar to LSTM implementation. It should also be noted that the inconsistency of results for a subset of 1 sensor may be largely due to specific road conditions (e.g. possible emergency), making it difficult to generalise results for the rest of the sensors.

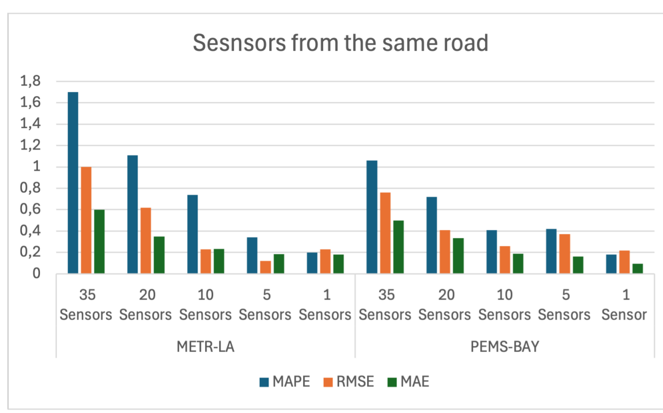

Figure 6: Difference in models' performance from sensors on the same road (red areas in Figures 2 and 3) for horizon 1.

A similar trend can be observed in Figure 7. Again, we can see a consistent decrease in the difference between LSTM and

DCRNN up to the amount of 10 sensors for both datasets. There is a slight decrease in the results for the METR-LA dataset with a subset of 5 sensors, where the difference in the RMSE metric was the lowest at 0.07. However, no decrease was observed for the same amount of sensors in the PEMS-BAY area, so the result may depend on the sensors selected and be rather site-specific. When only one sensor is considered, the differences are minimal for both road conditions, with one exception in Figure 7 for the METR-LA dataset, where only the difference in MAE metric decreased while the other two metrics surprisingly increased. However, again, for the reasons mentioned earlier, we should not generalise the result of this particular sensor to the rest of the sensors.

Contrary to initial expectations, the LSTM model performs more closely to the DCRNN model when the sensors are distributed on different roads, rather than on the same road. This might be because the DCRNN does not have enough sensor data to learn to build complex dependencies between sensors, especially given that sensors from the same road may be far apart in terms of road distance. This may limit its advantage over the LSTM. However, the DCRNN can still effectively use the same number of sensors placed on the same road.

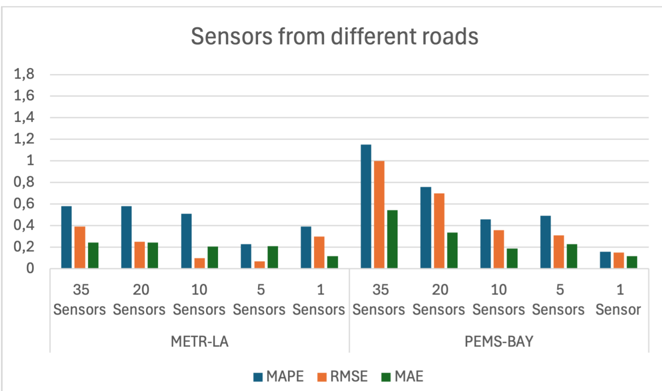

Figure 7: Difference in models' performance from sensors on different roads (blue areas in Figures 2 and 3) for horizon 1.

Experiments proved that with more sensors, the performance differences between DCRNN and LSTM increase. Nevertheless, when there are fewer sensors (10,5,1), the performance of LSTM was close to that of DCRNN. Furthermore, as mentioned earlier, the hyperparameters of the LSTM model were not tuned to be the most optimal, giving us room for improvement and perhaps outperforming DCRNN in these scenarios.

Analysis by road configuration

The scenario of different sensor deployment along roads has already been indirectly discussed in the two previous analyses. Interestingly, both indicate that the LSTM performs closer to the DCRNN in scenarios where sensors are distributed along different intersecting roads. This phenomenon is even more pronounced under the difficult Los Angeles road conditions. Here, the difference in metrics between the models increases only slightly as the forecast horizon extends (Figure 5), which is an exception compared to the other scenarios.

AVERAGE MAE DIFFERENCE		
Scenario	Same Road	Diff. Roads
METR-LA	0.3936	0.2506
PEMS-BAY	0.34481541	0.35537417

Table 3: MAE Average Difference for Same and Different Roads

Table 3 shows the average MAE error difference gathered across all 3 horizons and for 5 subsets of sensors on each road configuration. This data also reveals that the performance difference between the two models is the smallest for subsets of sensors from different roads in METR-LA. Interestingly, this is also where the models, on average, performed the worst. On the other hand, for the PEMS-BAY dataset, the placement of the sensors on the roads does not significantly impact the performance difference between the models. This finding partly contradicts the initial hypothesis that the difference in performance between LSTM and GNN should be smaller in situations where the sensors are positioned on the same road without many intersections. Furthermore, the LSTM even slightly outperformed the GNN model in the RSME metric in a subset of 10 sensors from the METR-LA dataset distributed over 3 different highways and a forecast horizon of 3 (Table 5 in the appendix A). The models scored 6.04 for DCRNN and 5.99 for LSTM in the aforementioned metric.

Analysis of the Training Time

To compare the time performance of both models, we analysed the average epoch learning times for the DCRNN and LSTM models (with 256 and 50 LSTM units) in 3 subsets of sensors coming from the different roads (blue area in Figure 2). We decided to include the performance of the LSTM model with only 50 units to better understand what affects the time performance of LSTM. The results are summarised in Figure 8. As expected, the LSTM models showed the shortest epoch learning times in all subsets. It is worth noting that the training time of the LSTM models is more influenced by the number of units in the model rather than the number of sensors used in training. This represents a significant advantage over DCRNN, where the average training time increases considerably with the number of sensors in the subset.

Another key factor to consider when discussing learning times is the number of epochs required for the models to converge. Figure 9 illustrates the training curves for these models on a subset of 35 sensors. Although DCRNN requires significantly more time (155.5 seconds) to train a single epoch, it converges faster in terms of the number of epochs than the other models. This is particularly evident when we compare this with the number of epochs that the model with 50 LAST units needed to converge.

Consequently, the actual training times of these models up to the early stopping point were as follows:

- DCRNN: 22 minutes and 48 seconds
- LSTM with 256 units: 8 minutes and 46 seconds
- LSTM with 50 units: 7 minutes and 9 seconds

These findings show that, while LSTM models offer shorter epoch training times, the faster epoch convergence rate of DCRNN can sometimes offset the longer training time per epoch, reducing the difference in overall training time for smaller subsets between DCRNN and LSTM. It should be noted, however, that the learning behaviour of the models may vary across different subsets of sensors, making it difficult to generalise the training curve to different scenarios.

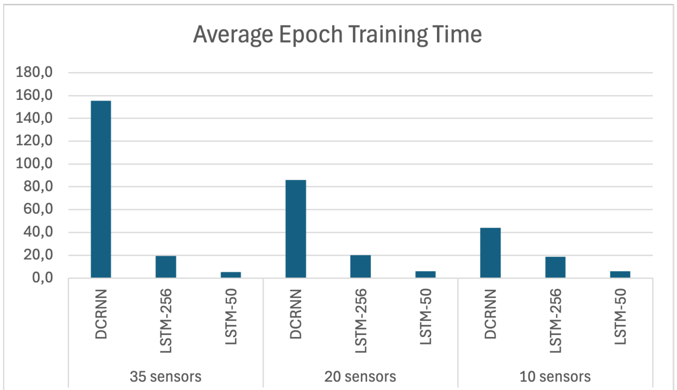

Figure 8: Average epoch training times (in seconds) for DCRNN and LSTM models (with 256 and 50 units) across three subsets of sensors.

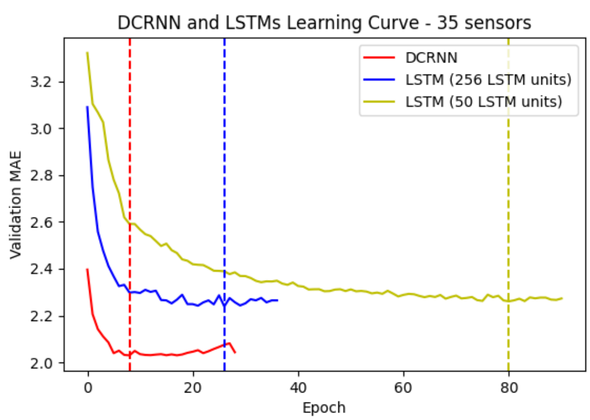

Figure 9: Training Curve of 3 models on the subset of 35 METR-LA sensors taken from different roads (blue area in Figure 2). The vertical lines indicate early stopping epochs.

Summary of Findings

DCRNN: Consistently outperformed LSTM under scenarios with a higher number of sensors and longer prediction horizons.

LSTM: Performed comparably with DCRNN in a scenario where the number of sensors was reduced and for the closest prediction horizon. More precisely, for the scenarios with 10, 5 and 1 sensors, the performance difference between DCRNN and LSTM was very small. This can be considered as promising results as there is a big room for improvement for LSTM which can even increase its performance.

Impact of Sensor Distribution: Interestingly, the LSTM performed closer to the DCRNN when the sensors were placed along different intersecting roads rather than on the same road. This was contrary to the initial hypothesis and suggests that GNN models have difficulty handling complex road configurations with limited sensor availability.

Training times: The analysis showed that DCRNN training time is very sensitive to the number of sensors, increasing rapidly as their number increases. Despite longer times per epoch, faster DCRNN epoch convergence can reduce the overall learning time difference. On the other hand, the training time of the LSTM depends strongly on the number of LSTM units used in a model. However, once the optimal configuration of hyperparameters is found, LSTM has a significant advantage over DCRNN in terms of training time.

4 Responsible Research

Ethical issues should be kept in mind when doing research using data. In this study, one potential ethical issue was the risk of identifying specific individuals using the data, as traffic data can sometimes reveal patterns that can be traced back to individual behaviour. Since the data was aggregated at 5-minute intervals, they were anonymised.

To ensure the reproducibility of the results, the models with their hyperparameters and the experimental procedure are clearly described in the paper. All datasets are publicly available, as are the subsets of sensors used in the experiments. In addition, all experimental results are available in the Tables 6 and 5 in the appendix A. Given that these were the only potential ethical concerns identified, there is no need to address further ethical issues in this study.

5 Conclusion

This study aimed to investigate scenarios in which simpler neural network models, in particular LSTMs, can perform comparably to complex GNNs, such as DCRNN, in traffic forecasting tasks. We have shown that while DCRNN provides higher accuracy in most traffic scenarios, LSTM can be an effective alternative under certain conditions. In particular, LSTM performed comparably to DCRNN in scenarios with 10, 5 and 1 sensors and the shortest prediction horizon. Interestingly, the results obtained only partially confirm the initial hypothesis. They indicate that with a limited number of sensors, the difference in performance was smaller. However, this was not necessarily the case in scenarios where they were distributed along a single road, as initially assumed.

6 Future Work

- **Hyperparameter Optimization**: The paper explores the capabilities of GNN and a simpler model in traffic forecasting. As noted, we did not use the most optimal version of the LSTM model. Consequently, one important improvement would be to optimise the hyperparameters of the LSTM models to improve their performance. Tuning parameters such as the number of layers, LSTM units and dropout rates could potentially close the DCRNN performance gap.
- **Extended Traffic Scenarios**: Expanding the range of traffic scenarios, including different types of road networks (e.g. urban areas) or other types of input data (e.g.

traffic volumes) [4]. This can provide a better understanding of the conditions under which simpler models may be most effective.

- **Comparative Studies with Other Models**: Due to time and resource constraints, only two models were tested in the experiments. Therefore, future research should also consider comparing other models emerging in the field of traffic forecasting, such as current and previous state-of-the-art models: Spatio-Temporal Graph Mixformer [5] or Fully Connected Long-Short Term Memory networks [12].

References

[1] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014.

[2] 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.

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

[4] Weiwei Jiang and Jiayun Luo. Graph neural network for traffic forecasting: A survey. *CoRR*, abs/2101.11174, 2021.

[5] Mourad Lablack and Yanming Shen. Spatio-temporal graph mixformer for traffic forecasting. *Expert Systems with Applications*, 228:120281, 2023.

[6] Yaguang Li, Rose Yu, Cyrus Shahabi, and Yan Liu. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *International Conference on Learning Representations (ICLR)*, 2018.

[7] Xu Liu, Yuxuan Liang, Chao Huang, Hengchang Hu, Yushi Cao, Bryan Hooi, and Roger Zimmermann. Do we really need graph neural networks for traffic forecasting?, 2023.

[8] Eric L. Manibardo, Ibai Laña, and Javier Del Ser. Deep learning for road traffic forecasting: Does it make a difference? *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6164–6188, 2022.

[9] Jesús Mena-Oreja and J. Gozalvez. A comprehensive evaluation of deep learning-based techniques for traffic prediction. *IEEE Access*, PP:1–1, 01 2020.

[10] Ramezani Saeed. Revisiting dcrnn: Diffusion convolutional recurrent neural network - data-driven traffic forecasting. https://medium.com/@saeedrmd/revisiting-dcrnn-diffusion-convolutional-recurrent-neural-network-data-driven-traffic-forecasting-caeecbe3281b, 2023.

[11] Zezhi Shao, Zhao Zhang, Fei Wang, Wei Wei, and Yongjun Xu. Spatial-temporal identity: A simple yet effective baseline for multivariate time series forecasting, 2022.

[12] Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

[13] David Tedjopurnomo, Zhifeng Bao, Baihua Zheng, Farhana Choudhury, and A. Qin. A survey on modern deep neural network for traffic prediction: Trends, methods and challenges. *IEEE Transactions on Knowledge and Data Engineering*, PP:1–1, 06 2020.

[14] Yongxue Tian and Li Pan. Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity)*, pages 153–158, 2015.

[15] Mingrong Tong and Hengxin Xue. Highway traffic volume forecasting based on seasonal arima model. *Journal of Highway and Transportation Research and Development (English Edition)*, 3:109–112, 12 2008.

[16] Juan José Vázquez, Jamie Arjona, MªPaz Linares, and Josep Casanovas-Garcia. A comparison of deep learning methods for urban traffic forecasting using floating car data. *Transportation Research Procedia*, 47:195–202, 2020. 22nd EURO Working Group on Transportation Meeting, EWGT 2019, 18th – 20th September 2019, Barcelona, Spain.

[17] Bing Yu, Haoteng Yin, and Zhanxing Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence*, IJCAI-2018. International Joint Conferences on Artificial Intelligence Organization, July 2018.

Appendix

A Experiment Result Tables

	MAE		RMSE		MAPE	
	DCRNN	LSTM	DCRNN	LSTM	DCRNN	LSTM
METR-LA						
Same Road	2.3487	2.7538	0.0551	0.0659	6.2820	6.8640
Diff Roads	3.0300	3.2760	0.0739	0.0810	6.8920	7.1247
PeMS-BAY						
Same Road	1.1367	1.4807	0.0222	0.0298	2.3113	2.9240
Diff Roads	1.2082	1.5814	0.0165	0.0222	1.7460	2.2240

Table 4: Average performance results of DCRNN and LSTM models for different road scenarios.

	MAE		MAPE		RMSE		Difference		
	DCRNN	LSTM	DCRNN	LSTM	DCRNN	LSTM	MAE	MAPE	RMSE
SAME ROAD 35									
Horizon 1	2.0500	2.6500	0.0549	0.0719	3.7400	4.7400	0.6000	0.0170	1.0000
Horizon 2	2.4700	3.2070	0.0693	0.0897	4.9100	6.2400	0.7370	0.0204	1.3300
Horizon 3	2.7900	3.6790	0.0821	0.1046	5.8000	7.4500	0.8890	0.0225	1.6500
SAME ROAD 20									
Horizon 1	1.9300	2.2780	0.0512	0.0623	3.6400	4.2600	0.3480	0.0111	0.6200
Horizon 2	2.2800	2.7364	0.0641	0.0790	4.8500	5.6000	0.4560	0.0149	0.7500
Horizon 3	2.5500	3.0728	0.0743	0.0923	5.7200	6.5500	0.5230	0.0180	0.8300
SAME ROAD 10									
Horizon 1	2.0400	2.2742	0.0432	0.0506	5.6400	5.8700	0.2340	0.0074	0.2300
Horizon 2	2.5300	2.8653	0.0532	0.0639	7.2500	7.5800	0.3350	0.0107	0.3300
Horizon 3	2.9100	3.3126	0.0608	0.0735	8.3600	8.7500	0.4030	0.0127	0.3900
SAME ROAD 5									
Horizon 1	1.9200	2.1050	0.0417	0.0451	5.7800	5.9000	0.1850	0.0034	0.1200
Horizon 2	2.4100	2.6883	0.0512	0.0568	7.4900	7.7200	0.2780	0.0056	0.2300
Horizon 3	2.8200	3.1784	0.0586	0.0661	8.7000	9.0300	0.3580	0.0075	0.3300
SAME ROAD 1									
Horizon 1	1.7000	1.8800	0.0333	0.0353	5.7900	6.0200	0.1800	0.0020	0.2300
Horizon 2	2.1900	2.4462	0.0405	0.0448	7.6000	7.9300	0.2560	0.0043	0.3300
Horizon 3	2.6400	2.9342	0.0476	0.0529	8.9600	9.3200	0.2940	0.0053	0.3600
DIFFERENT ROADS 35									
Horizon 1	1.9700	2.1490	0.0402	0.0460	3.4500	3.8400	0.2450	0.0058	0.3900
Horizon 2	2.1900	2.4660	0.0474	0.0545	4.2100	4.6300	0.2760	0.0071	0.4200
Horizon 3	2.3300	2.6379	0.0525	0.0600	4.7000	5.2000	0.3080	0.0075	0.5000
DIFFERENT ROADS 20									
Horizon 1	2.1500	2.3949	0.0407	0.0465	6.1400	6.3900	0.2450	0.0058	0.2500
Horizon 2	2.6100	2.9179	0.0486	0.0560	7.9100	8.1400	0.3080	0.0074	0.2300
Horizon 3	2.9800	3.3205	0.0544	0.0629	9.0700	9.3300	0.3410	0.0085	0.2600
DIFFERENT ROADS 10									
Horizon 1	2.4800	2.6869	0.0556	0.0607	4.3000	4.4000	0.2070	0.0051	0.1000
Horizon 2	2.8600	3.0810	0.0672	0.0729	5.3600	5.3600	0.2210	0.0057	0.0000
Horizon 3	3.0900	3.3132	0.0749	0.0801	6.0400	5.9900	0.2230	0.0052	-0.0500
DIFFERENT ROADS 5									
Horizon 1	2.6700	2.8788	0.0648	0.0671	6.8600	6.9300	0.2090	0.0023	0.0700
Horizon 2	3.3600	3.6951	0.0811	0.0889	8.6800	9.0600	0.3350	0.0078	0.2000
Horizon 3	3.8500	4.2814	0.0909	0.1035	10.2000	10.5300	0.4310	0.01260	0.3300
DIFFERENT ROADS 1									
Horizon 1	3.4600	3.5765	0.1071	0.1110	6.8700	7.1700	0.1170	0.0039	0.3000
Horizon 2	4.3800	4.4770	0.1318	0.1415	8.9500	9.1800	0.0970	0.0097	0.2300
Horizon 3	5.0700	5.1973	0.1507	0.1640	10.4600	10.7200	0.1270	0.0133	0.2600

Table 5: All performance results of DCRNN and LSTM models for different road scenarios on METR-LA dataset

	MAE		MAPE		RMSE		Difference		
	DCRNN	LSTM	DCRNN	LSTM	DCRNN	LSTM	MAE	MAPE	RMSE
SAME ROAD 35									
Horizon 1	0.9700	1.4702	0.0187	0.0293	1.7600	2.5200	0.5002	0.0106	0.7600
Horizon 2	1.2300	1.8129	0.0244	0.0368	2.3300	3.3500	0.5829	0.0124	1.0200
Horizon 3	1.4300	2.0568	0.0290	0.0424	2.8600	4.0000	0.6268	0.0134	1.1400
SAME ROAD 20									
Horizon 1	1.0800	1.4142	0.0207	0.0279	1.9900	2.4000	0.3342	0.0072	0.4100
Horizon 2	1.3500	1.8103	0.0267	0.0363	2.5600	3.2200	0.4603	0.0096	0.6600
Horizon 3	1.5500	2.0814	0.0313	0.0423	3.0200	3.8600	0.5314	0.0110	0.8400
SAME ROAD 10									
Horizon 1	0.8800	1.0688	0.0162	0.0203	1.5700	1.8300	0.1868	0.0041	0.2600
Horizon 2	1.1000	1.3956	0.0207	0.0276	2.1200	2.6500	0.2956	0.0069	0.5300
Horizon 3	1.2500	1.6436	0.0241	0.0336	2.6200	3.3600	0.3936	0.0095	0.7000
SAME ROAD 5									
Horizon 1	0.7700	0.9323	0.0136	0.0178	1.4200	1.7900	0.1623	0.0042	0.3700
Horizon 2	0.9700	1.2136	0.0179	0.0241	2.0100	2.6300	0.2436	0.0062	0.6200
Horizon 3	1.1400	1.4358	0.0220	0.0295	2.6000	3.3400	0.2958	0.0075	0.7400
SAME ROAD 1									
Horizon 1	0.8400	0.9357	0.0163	0.0181	1.7400	1.9600	0.0957	0.0018	0.2200
Horizon 2	1.1200	1.3180	0.0224	0.0269	2.6900	3.0300	0.1980	0.0045	0.4300
Horizon 3	1.3700	1.6238	0.0283	0.0342	3.4700	3.9600	0.2538	0.0059	0.4900
DIFFERENT ROADS 35									
Horizon 1	0.8600	1.4024	0.0164	0.0279	1.5400	2.5400	0.5424	0.0115	1.0000
Horizon 2	1.1700	1.7354	0.0230	0.0353	2.3500	3.4000	0.5654	0.0123	1.0500
Horizon 3	1.3900	1.9786	0.0284	0.0409	3.0400	4.0800	0.5886	0.0125	1.0400
DIFFERENT ROADS 20									
Horizon 1	0.9200	1.2562	0.0175	0.0251	1.6500	2.3500	0.3362	0.0076	0.7000
Horizon 2	1.2100	1.6240	0.0241	0.0329	2.4500	3.2800	0.4140	0.0088	0.8300
Horizon 3	1.4300	1.8843	0.0292	0.0386	3.1000	4.0100	0.4543	0.0094	0.9100
DIFFERENT ROADS 10									
Horizon 1	1.0200	1.2074	0.0197	0.0243	1.8100	2.1700	0.1874	0.0046	0.3600
Horizon 2	1.3400	1.6312	0.0268	0.0332	2.6000	3.1400	0.2912	0.0064	0.5400
Horizon 3	1.5900	1.9357	0.0327	0.0398	3.3100	3.9200	0.3457	0.0071	0.6100
DIFFERENT ROADS 5									
Horizon 1	1.2600	1.4900	0.0231	0.0280	2.1700	2.4800	0.2271	0.0049	0.3100
Horizon 2	1.6600	2.0300	0.0318	0.0390	2.9700	3.5200	0.3739	0.0072	0.5500
Horizon 3	1.9400	2.4156	0.0385	0.0471	3.6500	4.3300	0.4756	0.0086	0.6800
DIFFERENT ROADS 1									
Horizon 1	1.4300	1.5491	0.0233	0.0249	2.3400	2.4900	0.1191	0.0016	0.1500
Horizon 2	1.7400	1.9333	0.0284	0.0312	2.8000	3.0800	0.1933	0.0028	0.2800
Horizon 3	1.9300	2.1512	0.0316	0.0349	3.1000	3.4400	0.2212	0.0033	0.3400

Table 6: All performance results of DCRNN and LSTM models for different road scenarios in PeMS-BAY dataset.

B ChatGPT Usage

ChatGPT was mainly used for LaTeX formatting, fixing bugs in the code and helping to understand topics.

- **Prompt:** Given the H5 file, I am trying to use this (H5) file but got this error: raise ValueError(ValueError: Dataset(s) incompatible with Pandas data types, not table, or no datasets found in HDF5 file. could you redefined it to fix it?
- **Promt:** https://medium.com/@saeedrmd/revisiting-dcrnn-diffusion-convolutional-recurrent-neural-network-data-driven-traffic-forecasting-caeecbe3281b Could you Cite it using BibTeX format
- **Prompt:** adj_mx = np.exp(-np.square(dist_mx / std)) what to do if std is 0
- **Prompt:** what is diffusion process and diffusion convolution technique
- **Prompt:** Given this table (a screenshot of the table from my Excel spreadsheet), could you convert it into a LaTeX format
- **Prompt:** how to add a page counter in LaTex
- **Promt:** Should I use acronyms or full names of the models in the conclusion
- **Promt:** how to refer to appendix in LaTeX
- **Prompt:** _logger.info("val: " + val_loss + ", epoch: " + epoch + ", " + (end_time - start_time)) could you fix the format
- **Promt:** how to refer to appendix in LaTeX
- **Promt:** in which publication ARIMA was introduced first time
- **Promt:** can you keep all brackets the same size in the equation in LaTeX