Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2019) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

22nd EURO Working Group on Transportation Meeting, EWGT 2019, 18-20 September 2019, Barcelona, Spain

A Comparison of Deep Learning Methods for Urban Traffic Forecasting using Floating Car Data

Juan José Vázquez[a,*], Jamie Arjona[a], M[a]Paz Linares[a], Josep Casanovas-Garcia[a,b]

[a]*Universitat Politècnica de Catalunya, Barcelona 08034, Spain*
[b]*Barcelona Supercomputing Center, Barcelona 08034, Spain*

Abstract

Cities today must address the challenge of sustainable mobility, and traffic state forecasting plays a key role in mitigating traffic congestion in urban areas. For example, predicting path travel time is a crucial issue in navigation and route planning applications. Furthermore, the pervasive penetration of information and communication technologies makes floating car data an important source of real-time data for intelligent transportation system applications. This paper deals with the problem of forecasting urban traffic when floating car data is available. A comparison of four deep learning methods is presented to demonstrate the capabilities of the neural network approaches (recurrent and/or convolutional) in solving the traffic forecasting problem in an urban context. Different tests are proposed in order to not only evaluate the developed deep learning models, but also to analyze how the penetration rates of floating cars affect forecasting accuracy. The presented experiments were designed according to a microscopic traffic simulation approach in order to emulate floating car data fleets, which provide vehicle position and speed, and to validate the obtained results. Finally, some conclusions and further research are presented.

Keywords: urban traffic forecast; deep learning; floating car data.

1. Introduction

Traffic forecasting has been an active research topic since the late 1970s. It is far too general of a problem and includes different sub-problems with different degrees of complexity, depending on some aspects such as context, data source, predicted variables, and the prediction horizon, among others. This research focuses on traffic forecasting in urban contexts using floating car data (FCD) to predict the average speed of the roads on the network.

* Corresponding author. Tel.:+34934011019
E-mail address: juan.jose.vazquez.gimenez@fib.upc.edu

The existing literature presents many different approaches to solving the forecasting problem, which van Lint et al. (2012) classifies as the following: naïve methods that make no model assumptions; parametric methods whose structures are predetermined according to theoretical considerations that fit the parameters with data; and non-parametric methods whose structures and parameter values are determined from data.

Due to increases in the quantity and different sources of data, as well as the computational capabilities of new systems, the trend in recent years has changed in favor of non-parametric methods, specifically machine learning methods. In the field of traffic forecasting, deep learning (DL) proposals have proven to give more accurate predictions, and the use of these methods have increased intensively in recent years. For this reason, our work here focuses on developing DL based models.

This contribution is organized as follows. First, Section 2 summarizes a literature review of traffic forecasting using DL methods. Then, Section 3 specifies the selected methods to be compared. Section 4 presents the computational experiments by detailing the simulation scenarios, the proposed experimental design, the hyper-parameter optimization and the obtained results. Finally, Section 5 describes the final conclusions and some future research.

2. Related work

Vlahogianni et al. (2014) and Lana et al. (2018) systematically examine recent developments in data-driven traffic forecasting methods. The evolution between these two works shows a clear increase in the use of non-parametric methods in this field. In particular, DL is the most salient and recommended approach in recent proposals.

Long short-term memory (LSTM) methods are some of the most used DL approaches in studies on time series and other sequential data such as traffic data. These methods are considered a subfamily of recurrent neural networks (RNN) and are able to learn long-term dependencies while remembering information for long periods. The proposals of Duan et al. (2016), Liu et al. (2017), and Du et al. (2018) are good examples of applying LSTM to traffic forecasting. Fu et al. (2017) compare an LSTM model with a gated recurrent units (GRU) model, which is an NN method similar to LSTM and suggested by Cho et al. (2014). The results conclude that GRU outperforms LSTM in traffic forecasting.

In order to add spatial information to the previous LSTM methods, some authors propose merging LSTM with convolutional neural networks (CNN). In this way, Yu et al. (2017) present a spatiotemporal recurrent convolutional networks model (SRCN), which take as input a set of static images that represents the network-wide traffic speeds. Moreover, Cheng et al. (2017) introduce an end-to-end framework called DeepTransport, in which CNN and RNN are utilized to obtain spatial-temporal traffic. In addition, Cui et al. (2018) propose an approach that merges CNN and LSTM, which they call a High-Order Graph Convolutional Long Short-Term Memory Neural Network (HGC-LSTM). This applies CNN to the network graph encoded as a matrix, which is a similar format to images. The experiments presented in these papers demonstrate that the methods capture the complex relationships in the spatiotemporal domain and outperform traditional state-of-the-art DL methods.

Due to the lack of a generic suite for testing these kinds of solutions under the same conditions, it is difficult to compare different methods using the results from the original papers. Therefore, our proposal implements four of the most relevant proposals introduced above in order to compare them properly. In particular, we consider two recurrent neural network approaches (LSTM and GRU), and two of the previously mentioned combined solutions, SRCN and the HGC-LSTM methods.

3. Selected models

As indicated in the previous section, we implemented four different DL methods in order to perform traffic forecasting in urban contexts, using FCD to predict the average speed of the network road sections. Before delving into the different methods, the common initial data format should be defined. A new dataset is generated from the source FCD in order to represent the state of the network in different time periods of a predefined duration tdp. The form of the new dataset S is $S \in \mathbb{R}^{NxM}$, where N is the number of time windows with duration tdp and M is the number of road sections. So, for a time period i and a section j, S_{ij} represents the average speed of all recorded vehicles in j during i. S_{ij} is a missing value when the original data has no records for a section j at time period i. Because the proposed models do not accept missing values, they are imputed by performing a k-nearest neighbor imputation.

The LSTM and GRU models can be defined as a sequence of one or more specific layers (LSTM and GRU, respectively), which are all connected to each other. The last layer of each model is a fully connected NN layer for transforming the output of the last layer to the desired format (in this case, one value for each predicted road section speed). Therefore, given the state of the network in a period (S_i), these models are able to predict the state of the network in the next time period (S_{i+1}).

For the SRCN method, the input data incorporates the spatial relationships of the data, and each input state is codified as an image. To achieve this, an image template is defined by mapping the network model to a grid, where each cell represents a pixel. Thus, to build the state image for a period, each pixel of the image is filled by computing the average speed of S for that period and for all the road sections of the corresponding cell.

The inputs of the HGC-LSTM model are also 2-dimensional. Although they are not images, they are numerical matrices that can be interpreted as images. Given a predefined value K, and for a time period t, the input is computed with the following equation: $TS_t = [TS_t^1, \dots, TS_t^K]$. Each element of the array is computed by $TS_t^k = (FFR \odot \tilde{A}^k) \cdot S_t$, where $\odot$ is the element-wise product of matrices. $FFR \in \mathbb{R}^{MxM}$ is a binary matrix where $FFR_{ij} = 1$ if a path exists from section i to section j in a time up to tdp. $\tilde{A}^k \in \mathbb{R}^{MxM}$ is also a binary matrix, where $\tilde{A}_{ij}^k = 1$ if a path exists from section i to section j and crossing exactly $k-1$ sections.

In contrast to the differences in the input generation process, the SRCN and the HGC-LSTM models share the same structure. Because of the new format of the input, these models combine the RNN layers with some extra CNN layers. The following four consecutive parts comprise their structure. (1) The first part of the model is a set of CNN layers, each one composed of a convolutional and a pooling layer. (2) A flatten layer transforms the output of this first part into the required RNN format. (3) In order to incorporate the temporal relation of data into the model, a set of RNN layers (LSTM or GRU) is added before the flatten one. (4) The last layer is a fully connected layer that transforms the output to the desired format. This structure allows the models to take images as inputs and extract spatial and temporal information from these inputs.

4. Computational experiments

Once the proposed methods are defined, we will expose the comparative methodology to evaluate them. Our work here proposes a traffic simulation approach to generating the needed FCD input. In contrast to using real FCD, generating data through simulation allows creating data for a great variety of scenarios in order to study the performance of models in different situations. In addition, this saves a lot of effort in terms of the time and cost required for collecting real data. That said, depending on the case, it is better to use real or simulated data. If the goal is to use the traffic forecasting model in a real scenario, real data is highly recommended. Otherwise, if the goal is to compare different methods and evaluate their general performance under different conditions, simulated data is a better option.

The source FCD is generated using Aimsun (2018), a microscopic traffic simulator able to model the interactions for each vehicle and also collect data from them individually. From the simulation, a record (vehicle identifier, speed, section, and lane) is collected for each connected vehicle in every pre-defined period. In the following proposed experiments, the collection period is 10 seconds.

4.1. Scenarios

In this study, we use two different urban traffic networks in Spain to evaluate the performance of the forecasting models: Camp Nou and Amara (see Fig. 1). The former represents a small area of Barcelona composed of 4 nodes and 22 sections. The latter, Amara, is a district in San Sebastian composed of 105 nodes and 192 sections. Using these urban scenarios allows us to analyze the performance of the forecasting models using different network features, such as size, capacity, and the topology of the roads, among others.

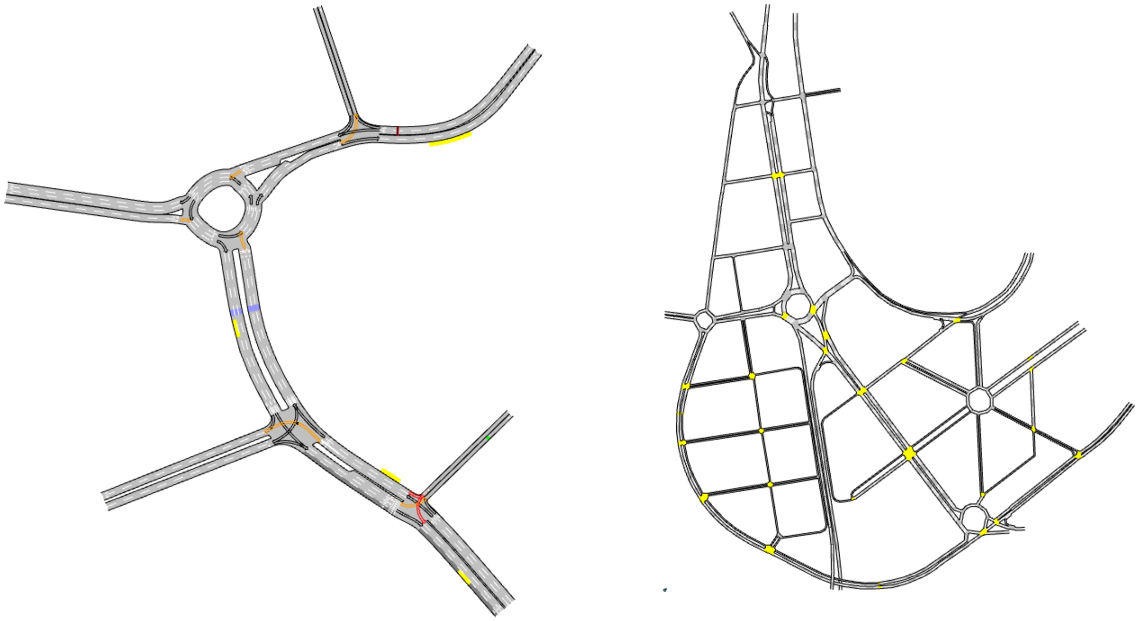

Fig. 1. Camp Nou (left) and Amara (Right) traffic simulation networks.

4.2. Experimental design

In order to evaluate the developed models, a partial factorial experimental design is defined. It is based on three experimental factors:

- Penetration rate: percentage of vehicles in the network used to generate the FCD. This factor is presented at four levels: 25%, 50%, 75%, and 100%.
- Prediction horizon: this defines how far ahead the model predicts the future. This factor is presented at six levels: 5, 10, 15, 20, 40 and 60 minutes. The first three levels are considered to be short-term, and the rest are long-term.
- Training data: amount of historical data used to train the methods. This is quantified by the number of days corresponding to the data. This factor is presented at three levels: 5, 10 and 15 days.

Every experiment is based on a common initial configuration set at a 100% penetration rate within a 5-minute prediction horizon and using 5 days' worth of data to train the models. For each experiment, the factor level is changed and the rest maintain their initial values. In addition, every selected factor configuration is tested for each of the four implemented models and in the two previously presented scenarios.

4.3. Hyper-parameter optimization

The selected deep learning methods require setting a group of parameters for use in an optimization process. This process is named hyper-parameter optimization. In particular, we optimize the hyper-parameters of the models for every test by using a random search as an alternative to grid search and manual search. The results presented by Bergstra & Yoshua (2012) show that this strategy is able to find models that are as good or better, and they also perform within a small fraction of the computation time needed by other search strategies. This algorithm consists of generating a random set of possible configurations and selecting the best one based on its accuracy with the validation dataset. In this case, 60 different random configurations are generated for each experiment.

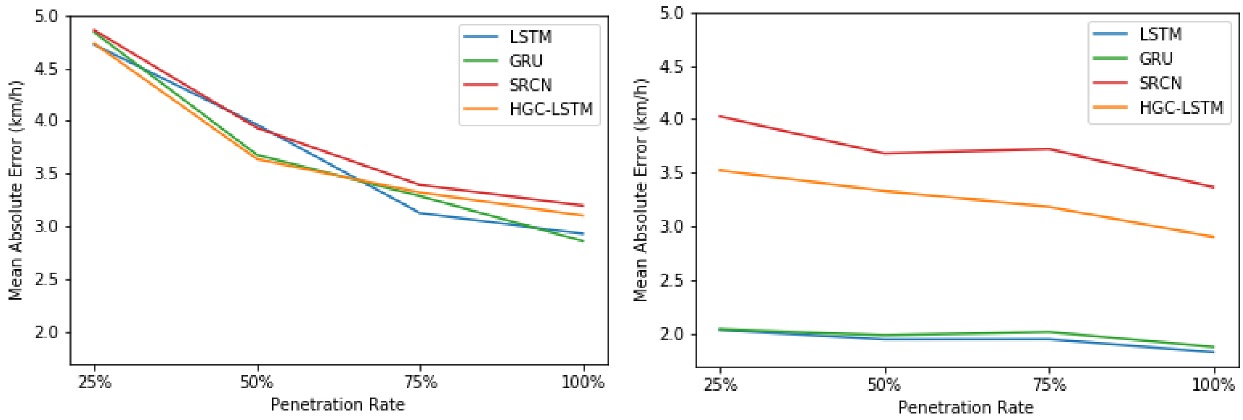

Fig. 2. MAE of the four models, depending on the FCD penetration ratio in Camp Nou (left) and Amara (right).

4.4. Results and discussion

In order to evaluate the performance of the implemented models, two different forecast error measures are used. A forecast error measure quantifies how well the forecasted values $\tilde{y} \in \mathbb{R}^N$ match the observed ones $y \in \mathbb{R}^N$. The error measures used to evaluate the models' accuracy are mean absolute error (Equation 1) and root mean squared error (Equation 2).

$$MAE = \frac{1}{N}\sum_i^N |y_i - \tilde{y}_i| \quad (1)$$

$$RMSE = \sqrt{\sum_i^N (y_i - \tilde{y}_i)^2} \quad (2)$$

The first experiments analyze the performance of the forecasting models using four different FCD penetration rates (25%, 50%, 75%, and 100%). The results presented in Table 1, Table 2 and Fig. 2 show that the best models for the highest penetration rates (100% and 75%) are LSTM and GRU. In the Amara scenario, which is the largest, these two models are also the best for the lowest penetration rates (50% and 25%). In contrast, for the Camp Nou scenario, the HGC-LSTM model is the best option for the lowest penetration values. Also, the decrease in the penetration rate directly affects prediction accuracy by causing it to also decrease, especially in the smallest scenario. Having said this, it is important to consider that the errors for the lowest penetration rate are reasonably good for forecasting urban traffic.

In general, the GRU and LSTM methods outperform the others in mostly all the short-term experiments performed for both the Camp Nou and Amara scenarios (see Tables 3 and 4). In the long-term case, the accuracy of the four options is very similar for the two scenarios. The prediction error of GRU and LSTM models is practically constant for long-term and short-term experiments, so the prediction horizon is not so critical for them. The performance of the convolutional methods is also very similar for all situations, with the exception of the smallest prediction horizons in the Amara scenario, where the results are worse than the others.

Table 1. Penetration rate – Camp Nou.

	25%		50%		75%		100%	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	4.72	8.227	3.963	6.134	3.128	5.196	2.932	4.865
GRU	4.837	8.318	3.677	5.817	3.287	5.292	2.862	4.776
SRCN	4.857	8.27	3.932	5.937	3.394	5.436	3.197	5.12
HGC−LSTM	4.731	8.306	3.637	5.836	3.322	5.408	3.104	5.03

Table 2. Penetration rate – Amara.

	25%		50%		75%		100%	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	2.04	4.948	1.951	4.834	1.953	4.835	1.831	4.602
GRU	2.045	5.185	1.991	5.016	2.021	5.062	1.88	4.73
SRCN	4.026	6.532	3.681	6.264	3.723	6.347	3.368	5.896
HGC−LSTM	3.525	6.034	3.332	5.91	3.186	5.811	2.905	5.412

Table 3. Prediction horizon – Camp Nou.

	Short-term						Long-term					
	5 min.		10 min.		15 min.		20 min.		40 min.		60 min.	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	2.931	4.864	2.917	4.839	2.879	4.814	2.991	4.913	2.953	4.863	3.063	4.922
GRU	2.862	4.775	2.868	4.767	2.873	4.769	2.912	4.804	2.892	4.802	2.927	4.874
SRCN	3.196	5.12	3.181	3.181	3.179	5.1	3.168	5.082	2.945	4.864	3.02	4.935
HGC−LSTM	3.103	5.029	3.179	5.11	3.322	5.178	3.164	5.084	3.2	5.077	3.185	5.061

Table 4. Prediction horizon – Amara.

	Short-term						Long-term					
	5 min.		10 min.		15 min.		20 min.		40 min.		60 min.	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	1.831	4.602	1.9	4.747	1.929	4.834	1.881	4.752	1.878	4.795	1.944	5.068
GRU	1.88	4.73	1.861	4.699	1.897	4.794	1.89	4.722	1.857	4.832	1.934	4.947
SRCN	3.363	5.891	2.477	5.076	1.851	4.791	1.849	4.811	1.848	4.84	1.837	4.859
HGC−LSTM	2.905	5.412	3.317	5.84	1.836	4.782	1.845	4.807	3.349	5.914	1.851	4.86

Lastly, the results of the training data experiments are presented (see Table 5). The methods that perform better, in general, are LSTM and GRU. In particular, GRU shows a smaller error for the Camp Nou network, and LSTM is the best option for the Amara scenario. The convolutional solutions are better for the Amara network in twice the number of cases, but the difference is small. Although the accuracy of the proposed forecasting methods generally increases with more data, the difference is not critical. This is especially true for the Amara network, where the improvement is minimal. Thus, the implemented models show good performance in traffic forecasting with 5 days of training data.

Table 5. Training data.

	Camp Nou						Amara					
	5 days		10 days		15 days		5 days		10 days		15 days	
Models	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
LSTM	2.931	4.864	2.852	4.645	2.69	4.606	1.812	4.097	1.412	3.085	1.565	3.55
GRU	2.862	4.775	2.718	4.492	2.749	4.628	1.799	4.057	1.607	3.537	1.567	3.524
SRCN	3.04	4.909	3.037	4.947	2.765	4.661	1.581	3.508	2.544	4.159	1.466	3.57
HGC−LSTM	3.103	5.029	2.889	4.676	2.783	4.689	1.581	3.508	3.401	5.349	1.469	3.578

5. Conclusions

This project deals with the traffic forecasting problem, which has been prominently active in the last 40 years. Traffic forecasting plays a key role in mitigating some traffic and transportation problems. In particular, this project focuses on traffic forecasting in urban networks using floating car data (FCD).

Four deep learning methods were implemented in order to perform traffic forecasting. The results of the performed experiments show that these solutions are able to predict traffic speeds with good performance. Specifically, recurrent methods (LSTM and GRU) present smaller errors than convolutional ones (SRCN and HGC-LSTM). Therefore, the convolutional component is not needed to extract spatial information.

In terms of penetration rate, its increment reduces the prediction error. However, the methods predict reasonably well with the smallest tested penetration (25%). Similarly, the use of more training data increases the accuracy, although the improvements are not very significant, and 5 days of data are enough to train the four tested methods.

Furthermore, the presented computational experiments determine that the implemented models are able to perform accurate traffic forecasting regardless of scenario size and prediction horizon. These results inspire further research to complement the performed experiments, such as extending the experimental design by adding more levels for the proposed factors as well as by considering new factors. Although the forecasting models in the literature usually test smaller scenarios than Amara, the use of a larger network could be interesting for evaluating the feasibility of the models in terms of their forecasting and computational capabilities. Furthermore, in order to perform more realistic predictions, differentiating section lanes could pose a highly interesting challenge in detecting traffic congestion.

FCD can at times be insufficient for covering all the network sections, and machine learning forecasting of a variable without any historical data does not make sense. Nevertheless, different approaches can be applied to solving this. For example, secondary methods may be used, the missing values can be extrapolated, or new data sources could be added. Aside from cases of missing values, including new data sources can complement the FCD and improve forecasting accuracy. The new data could be of the same type as that which we used here (speeds from loop sensors), or it could be completely different (exogenous variables like weather conditions or calendar events).

Acknowledgements

Throughout this work, the authors have benefited from the support of inLab FIB team at Universitat Politècnica de Catalunya. This research was funded by Secretaria d'Universitats i Recerca de la Generalitat de Catalunya (2017-SGR-1749) and under the Industrial Doctorate Program (2016-DI-79).

References

Aimsun. (2018). *Aimsun Next 8.3.0*. Retrieved from https://www.aimsun.com

Bergstra, J., & Yoshua, B. (2012). Random Search for Hyper-Parameter Optimization. *Journal of Machine Learning Research*. https://doi.org/10.1162/153244303322533223

Cheng, X., Zhang, R., Zhou, J., & Xu, W. (2017). DeepTransport: Learning Spatial-Temporal Dependency for Traffic Condition Forecasting. *CoRR*, *abs/1709.0*. Retrieved from http://arxiv.org/abs/1709.09585

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. *CoRR, abs/1406.1*. Retrieved from http://arxiv.org/abs/1406.1078

Cui, Z., Henrickson, K., Ke, R., & Wang, Y. (2018). *High-Order Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting*. Retrieved from http://arxiv.org/abs/1802.07007

Duan, Y., Lv, Y., Wang, F.-Y. (2016). Travel time prediction with LSTM neural network. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. https://doi.org/10.1109/ITSC.2016.7795686

Fu, R., Zhang, Z., & Li, L. (2017). Using LSTM and GRU neural network methods for traffic flow prediction. *Proceedings - 2016 31st Youth Academic Annual Conference of Chinese Association of Automation, YAC 2016*. https://doi.org/10.1109/YAC.2016.7804912

Lana, I., Del Ser, J., Velez, M., & Vlahogianni, E. I. (2018). Road Traffic Forecasting: Recent Advances and New Challenges. *IEEE Intelligent Transportation Systems Magazine*, *10*(2), 93–109. https://doi.org/10.1109/MITS.2018.2806634

Liu, Y., Wang, Y., Yang, X., & Zhang, L. (2017). Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models. *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. https://doi.org/10.1109/ITSC.2017.8317886 van Lint, H., & van Hinsbergen, C. (2012). Short-Term Traffic and Travel Time Prediction Models. In *Artificial Intelligence Applications to Critical Transportation Issues* (pp. 22–41). https://doi.org/10.17226/22690

Vlahogianni, E. I., Karlaftis, M. G., & Golias, J. C. (2014). Short-term traffic forecasting: Where we are and where we're going. *Transportation Research Part C: Emerging Technologies*, *43*, 3–19. https://doi.org/10.1016/j.trc.2014.01.005

Yu, H., Wu, Z., Wang, S., Wang, Y., & Ma, X. (2017). Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. *Sensors (Switzerland)*. https://doi.org/10.3390/s17071501