GANNSTER: Graph-Augmented Neural Network Spatio-Temporal Reasoner for Traffic Forecasting

Carlos Salort Sánchez[1,2], Alexander Wieder[1], Paolo Sottovia[1], Stefano Bortoli[1], Jan Baumbach[3], and Cristian Axenie[1]

[1] Huawei German Research Center, Munich, Germany
{carlos.salort, cristian.axenie}@huawei.com
[2] Technical University of Munich, Munich, Germany
[3] Chair of Experimental Bioinformatics, TUM School of Life Sciences Weihenstephan, Freising, Germany

Abstract. Traffic forecast is a problem of high interest due to its impact on mobility and inherent socio-economic aspects of people's lives. Particularly for adaptive traffic light systems, the ability to predict traffic throughput in intersections enables fast adaptation, thus reducing traffic jams. In this work, we propose a novel approach for traffic forecasting, termed Graph Augmented Neural Network Spatio-TEmporal Reasoner (GANNSTER), which fuses spatial information, given by the traffic network topology, with temporal reasoning and learning capabilities of recurrent neural networks. Our modelling contribution is supplemented by the public release of a novel real-world dataset containing urban traffic throughput in intersections. We comparatively evaluate GANNSTER against state-of-the-art models for traffic forecast and demonstrate its superior performance.

Keywords: Deep Learning · Graph Neural Network · Traffic Forecast

1 Introduction

Traffic congestion resulting from growing traffic volumes in urban areas has a major impact on life, ranging from socio-economic to environmental aspects, such as air pollution, commute time, and waste of energy. One key method for reducing congestion is to optimize the traffic light control accordingly based on the current traffic situation, but also the expected traffic in the near future. For instance, accurate traffic flow forecasts can be used to improve traffic light control, therefore reducing the formation of jams or minimizing their effects.

Forecasting traffic flow, however, constitutes a complex problem. Traffic depends on a large variety of factors, for instance, the length of the traffic light phases, the type of vehicles, the driver behaviour, and the variety of weather conditions. Additionally, datasets present high variability between consecutive readings from the road sensors. Such variability of consecutive measurements is illustrated with an example in Figure 1. Said measurements may be subject to

noise due to unforeseen external conditions (e.g. untrimmed trees, poor visibility at night, vandalism, etc.).

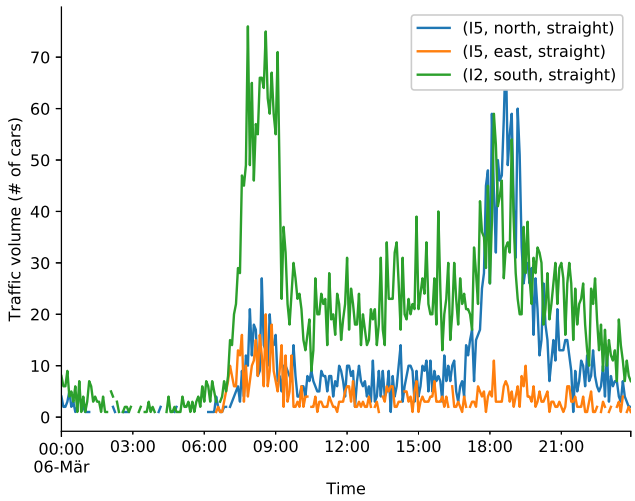

Fig. 1: Example for car throughput in an urban intersection. Data is aggregated every five minutes. Each line represents 24 hours of data for one traffic sensor.

A variety of different techniques for traffic forecasting have been proposed, ranging from statistical methods [1] to Deep Learning models [13,18]. Deep Learning (DL) methods have proved their potential in learning from large amounts of data. For instance, DL excels in describing sequences of temporally dependent values [4,12], hierarchical visual processing [15] and data generation [24]. While these studies present promising results in restricted scenarios, they fail to capture the spatio-temporal relationships in the data. In an attempt to introduce this dimension into the model, there has been a large amount of work addressing methods to embed graph structures into neural networks [2,7,10,14,28,33]. Some of this work has been applied in the traffic domain, but most of it focuses on forecasting the average speed of vehicles driving through detectors on highways rather than throughput of urban traffic.

In this work, we address the challenge of urban traffic flow forecasting, and introduce *GANNSTER: Graph Augmented Neural Network Spatio-TEmporal Reasoner*, a novel deep learning model and system that exploits both temporal and spatial information through embedded graphs for predicting traffic flow. The primary purpose of GANNSTER is to provide traffic flow predictions to be used for traffic light logic optimization. Since traffic light logic optimization requires predictions at most about one hour ahead, GANNSTER focuses on this short-term prediction horizon.

Our main contributions are:

- GANNSTER, a neural network-based system, that embeds and exploits the temporal and spatial relations in a road network for throughput prediction across intersections.
- The comparative evaluation of GANNSTER against relevant state-of-the-art models on a novel real-world dataset. Our evaluation results show that our approach generally yields higher accuracy than the other evaluated methods.
- A new real-world dataset, MUSTARD-S (Multi-cross Urban Signalized Traffic Aggregated Region Dataset - Small), which contains road traffic data recorded over 55 days and 6 intersections.

We start by describing related work employing recurrent neural networks, graph neural networks and other methods for traffic forecasting in Section 2. Following, in Section 3, we introduce GANNSTER. The experiments, along with their methodology, are introduced in Section 4. We present and discuss the evaluation and experimental results in Section 5. Finally, Section 6 concludes the paper and discusses opportunities for future work.

2 Related work

The proposed system taps into efficient solutions for traffic forecasting and explores how the new breed of graph neural networks can tackle the inherent dynamics of such a complex process. In the following, we provide an overview of key state-of-the-art approaches.

2.1 Traffic Forecasting

Most of the traffic forecasting approaches use statistical methods [1,9]. While these types of models work well on small datasets, the increasing number of traffic sensors, data quantities and heterogeneity, and computational power has made them obsolete. Recent approaches use machine learning techniques for detecting non-linear relations among the traffic variables. For instance, works using Support Vector Regression [25] tend to outperform statistical methods in terms of accuracy by learning a linear function in the space induced by a non-linear kernel which corresponds to a non-linear function in the original space. Following the trend from the machine learning community, researchers turned towards DL for traffic forecasting [22,26]. While outperforming both statistical and machine learning methods, base DL methods fail to exploit temporal and spatial information. With the rise of recurrent neural networks, and particularly Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) [4,12], researchers added the temporal dependency to the equation [20,31]. Yet, such systems still fail to capture the spatial information. In an attempt to add spatial information, some models incorporated convolutional neural networks [19,30,32], with the downside of not being able to accurately represent the road network topology.

2.2 Graph Neural Networks

Recent trends demonstrated an increasing interest in combining DL techniques and graphs. Whereas most data problems lie in a Euclidean space, that is not the case for graph data. Therefore some of the data assumptions do not hold (e.g. hierarchical representation, flatness, flexible operations with fewer dimensions).

To bypass these difficulties, some approaches included the use of graph-structured spatial information, such as the PATCHY-SAN that extracted locally connected regions from graphs using learned feature representations competitive with state-of-the-art graph kernels [21]. GraphSAGE is another approach that employs inductive learning to leverage graph node attribute information to efficiently generate representations of previously unseen data [10]. Another remarkable approach leverages Diffusion Convolution Networks that introduce a novel diffusion-convolution operation and diffusion-based representations that can be learned from graph-structured data and used as an effective basis for node classification [17]. Finally, Attention Based Methods stand out, particularly self-attention mechanisms, that relate different positions of a single sequence to compute a representation of the same sequence [27]. Among these, the method that has gained more popularity relies on generalizing convolutional neural networks [16] using spectral graph theory. This idea was first introduced in [3], with Graph Convolutional Networks (GCN). Several works [6,11,14] have been built on top of these principles using different approximations from spectral graph theory. These methods have been successfully applied to a growing set of problems, such as link prediction in optimizing networks [14], and representing three-dimensional protein structures [7]. A more detailed description of general GCN can be found in [28,33].

2.3 Graph Neural Networks for Traffic Forecasting

With the evolution of graph-based neural networks, a new opportunity to tackle traffic forecast problems arose. The work in [29] proposes Spatio-Temporal GCN (STGCN) to combine graph convolutions and gated temporal convolutions for extracting the most relevant spatial and temporal features coherently. By equipping a neural network with attention mechanisms, the work in [8] enabled focusing on a subset of inputs and features. Basically, by computing masks used to multiply features, the work proposes an attention mechanism with three independent temporal components, namely recent data, daily data, and weekly data, fused to generate the final traffic forecast. The work in [17] introduces Diffusion Convolutional Recurrent Neural Networks (DCRNN), a model employing bidirectional random walks on a graph to learn its spatial dependencies, and an encoder-decoder with scheduled sampling architecture to detect the temporal dependencies. Our method differs structurally from this approach, as we train the model directly, and propose an alternative convolution-based approach. Another important aspect is the fact that DCRNN uses a weighted graph based on the distance between sensors, which we avoid to reduce the amount of information

needed about the network. Finally, the work in [5] proposes Traffic Graph Convolutional LSTM (TGC-LSTM), a model based on LSTM, that uses Free-Flow Reachability (FFR) matrices in the graph convolution to provide extra information to the model. Similarly to our work, TGC-LSTM uses k-walks matrices, but without considering previous temporal values. Furthermore, our model does not depend on any extra information other than the adjacency matrix.

3 GANNSTER

In this section, we detail the structure of GANNSTER and the graph encoding of the road network to exploit spatio-temporal dependencies for traffic predictions. We start by introducing the road graph and definitions used throughout this section.

3.1 Road Graph

The road network structure is crucial for producing accurate traffic forecasts. For instance, knowing that a road segment is unidirectional often allows reasonably accurate predictions for the next intersection reached by a vehicle driving on that road. While this is intuitive for humans, such properties need to be carefully encoded into the model to enable forecasts.

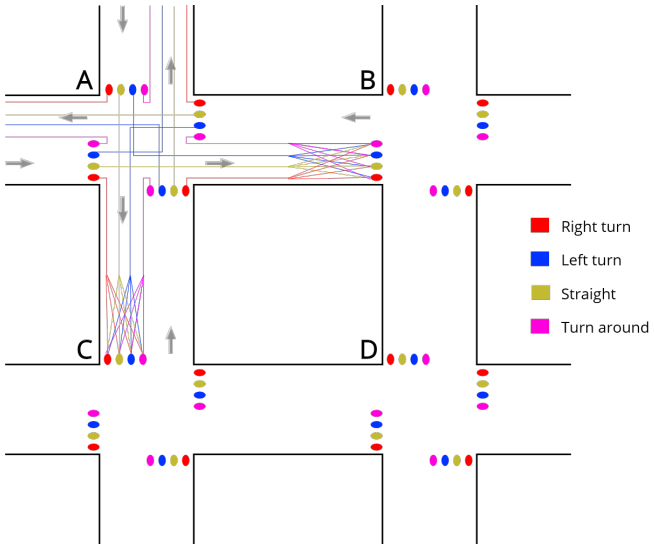

Fig. 2: Possible paths for a car entering intersection A. There are four possible directions, and four possible turns. As an example, a car entering (A, north, left) has four possible destinations: (B, west, {any direction}).

Figure 2 depicts an example of the graph derived from a map with four intersections. For the sake of readability, possible paths are only plotted for intersection A. Each intersection can be entered from four possible directions (north, east, south, and west), and can be left into four directions (by turning left, right, going straight, and turning around). Note that the topology in the depicted example was solely chosen for clarity, and other road structures can be encoded similarly. In the following subsection, we formally define the construction of the road graph.

3.2 Definitions

We define the road graph as a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where each vertex

$$v = (\text{intersection, source direction, turn}) \in \mathcal{V} \tag{1}$$

represents one concrete possibility for traversing an intersection (coming from a specific direction and taking a specific turn). We define that a directed edge

$$e = (v_{origin}, v_{destination}) \in \mathcal{E} \tag{2}$$

exists if and only if an intersection can be traversed as specified by $v_{destination}$ directly after traversing the same or a different intersection as specified by v_{origin}. We assume that sensors are installed at intersections counting vehicles for each possible source and turning direction (i.e., at each node in $\mathcal{G}$.). For brevity, we further define $N = |\mathcal{V}|$.

Throughout this work, we assume a discrete time model with equally-sized steps of five minutes. We define the *system state* at a specific time step t to consist of the the number of vehicles detected by each sensor since the previous time step, and we denote the system state at time t as $x^{(t)} \in \mathbb{R}^N$. We assume that, at any time, the recent *history* of system states is available, that is, the ordered sequence of system states from the T' most recent time steps is known. We define the *prediction horizon* as the number of steps (starting from the last known state) for which the system state shall be predicted, and denote it as T. Based on this notation, the forecasting problem can be phrased as the problem of finding a function h that satisfies

$$[x^{(t-T'+1)}, \ldots, x^{(t)}; \mathcal{G}] \xrightarrow{h} [x^{(t+1)}, \ldots, x^{(t+T)}]. \tag{3}$$

We let $A \in \mathbb{R}^{N \times N}$ denote the adjacency matrix of $\mathcal{G}$ defined in the common way, that is, $A_{i,j} = 1$ if $(v_i, v_j) \in \mathcal{E}$, 0 otherwise. Note that A is not necessarily symmetric since the edges in $\mathcal{G}$ are directed.

We define a *walk* as a sequence of edges $[e_1 = (v_0, v_1), \ldots, e_i = (v_{i-1}, v_i)]$, which connects a sequence of vertices in the graph. With M^k denoting the k'th power of a matrix M, given the adjacency matrix A of $\mathcal{G}$, the matrix A^k represents the number of possible walks of degree k. That is, $A^k_{i,j}$ represents the number of walks from vertex v_i to v_j with length k. In a road traffic graph, this

can be interpreted as the multitude of nodes that can be reached in k time steps by a vehicle originally detected in vertex v_i.

We define the *k-walk matrix* as

$$\hat{A}^k_{i,j} = min(A^k_{i,j}, 1), \tag{4}$$

such that $\hat{A}^k_{i,j} = 1$ if there is at least one k-degree walk from v_i to v_j, and $\hat{A}^k_{i,j} = 0$ otherwise. Each row and column of this matrix represents one vertex in the graph, and the matrix represents the final vertex (columns) where a vehicle can arrive starting from the initial vertex (row) in k steps. After multiplying $\hat{A}^k$ with the system state, we obtain $\hat{A}^k x^{t-k}$, a vector representing the maximum amount of cars that can arrive at a particular node from any node in the network in k steps.

We let $D^k \in \mathbb{R}^{N \times N}$ denote the *degree matrix* of $\hat{A}^k$: $D^k_{ii} = \sum_{j=1}^{N} \hat{A}^k_{i,j}$. This diagonal matrix represents the number of edges that can be reached in exactly k steps starting from the vertex v_i. The inverse of the degree matrix is represented by D^{-k}.

The GANNSTER model utilizes a similar approach to the *Graph Convolution* operation defined in [11,14], using the adjacency matrix as $\hat{A}x^{(t)}$ to extract local information from previous steps.

3.3 GANNSTER Model

GANNSTER incorporates both temporal and spatial information by leveraging a combination of Graph Convolutions and Recurrent Neural Networks (RNN).

Temporal Information GANNSTER utilizes RNNs, an established type of DL structures designed for use with temporal data. GANNSTER is agnostic to concrete type of RNN, and in this work, we instantiate GANNSTER in combination with LSTM and GRU. RNNs are well-suited for processing sequence data for predictions but suffer from short-term memory. LSTMs and GRUs mitigate short-term memory using gates that regulate the flow of information flowing through the sequence chain. In addition to the temporal traffic information, in either case, we augment the input vector for the t-step with additional information from the graph representing the road network.

Spatio-Temporal Information Road topology contains rich implicit information (e.g. adjacency, connectivity, directions). Our objective is to incorporate this spatio-temporal information into the RNN components of GANNSTER.

In Figure 3 we can see a vehicle, currently positioned in (A, west, straight), at time t. Let's assume that, in one timestep, it can move one intersection. Then, due to the road topology, we know that the vehicle will be in (C, west, any direction) at time $t + 1$. Furthermore, at time $t + 2$ the vehicle can be in (A,

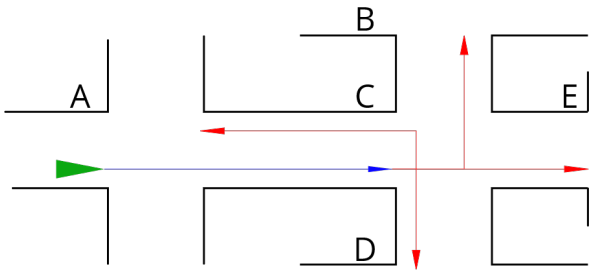

Fig. 3: Vehicle located in (A, west, straight), and possible paths in one timestep (blue) and in two timesteps (red).

east, any), (B, south, any), (D, north, any) or (E, west, any). This information can be used by the model to improve forecast accuracy.

We will use the matrix $\hat{A}^k$ to incorporate spatio-temporal information. As it represents the possible k-walks, when computing the product $\hat{A}^k x^{(t-k)}$, we obtain the number of vehicles from k timesteps ago and k hops away from i at position i. This represents a rich new source of information that constitutes the base for our model.

GANNSTER Network GANNSTER embeds the graph structure along with the temporal information into the model. We define the parameter K as the number of past steps that will be considered in the model.

We define a GANNSTER vector as

$$\text{GANNSTER}^t = \Big\|_{k=0}^{K} (D^{-k}\hat{A}^k x^{(t-k)}) \tag{5}$$

where $\|$ represents vector concatenation. The vector described in equation 5 will be the input of the RNN. In cases where $(t-k) < (t-T'+1)$, i.e. the input information for the model is not available because it is too old, we use $x^{(t-k)} = 0$. We use D^{-k} to normalize the number of cars in previous steps.

Figure 4 shows the architecture used by GANNSTER, when used jointly with LSTM. As explained before, other RNN structures can be used. We use many to many sequence prediction. The main difference with plain RNN architectures is the addition of the spatio-temporal information as input. Please note that, for the case $K = 0$, GANNSTERLSTM becomes a normal LSTM model. Analogous to other RNN-based systems, we can stack L blocks. We explored large scale structures and added a dropout layer between blocks. We only use GANNSTER vectors in the first block. In posterior blocks, the hidden space dimension does not necessarily match the input space dimension and therefore the k-walk matrix loses its meaning.

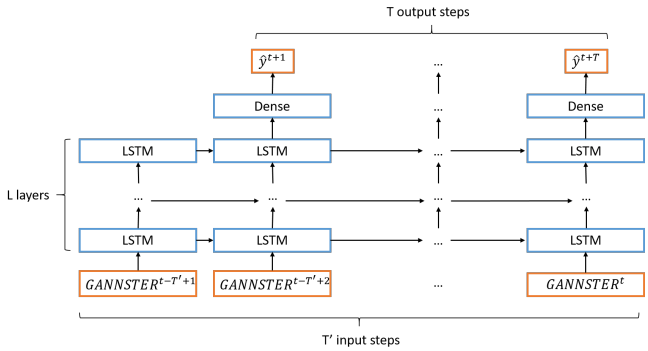

Fig. 4: GANNSTER architecture.

4 Experimental Evaluation

We conducted an experimental evaluation of GANNSTER to assess the performance in terms of forecasting accuracy. For comparison, we included state-of-the-art models that incorporate topological information about the road network encoded as graph, as well as simpler baseline models in our evaluation. In contrast to GANNSTER and the state-of-the-art methods, the baseline methods are oblivious to the structure of the road network, and hence, the comparison to them may indicate the performance benefit stemming from the additional topological information. For evaluating the performance in an urban setting, we introduce a novel dataset, MUSTARD-S (Multi-cross Urban Signalized Traffic Aggregated Region Dataset - Small), which we describe next.

4.1 MUSTARD-S

We present MUSTARD-S (Multi-cross Urban Signalized Traffic Aggregated Region Dataset - Small), a dataset consisting of 55 days of traffic throughput at six intersections in a city in China. We are working to increase the size of the dataset, and will be made public once available. The road network underlying this dataset is depicted in Figure 5.

4.2 Experimental Settings

For training, we use a 80/10/10 train/validation/test split. Due to the time dependency of the data samples, we chose a sequential split. For all models, we train for up to 400 epochs, with an initial learning rate of 0.0001, and Mean Squared Error (MSE) as a loss function. We have a patience mechanism for updating the learning rate. Once the validation error does not improve by at least 0.00001 for 10 iterations, we decrease the learning rate by a factor of 10, resort to the iteration that achieved the best accuracy in validation and resume training from that state on. We stop training after the learning rate was updated

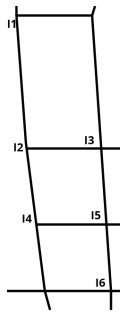

Fig. 5: MUSTARD-S map. Named intersections are considered in the study.

twice, or the epoch limit is reached. We normalize the dataset using Z-score. History size is one hour of data (12 data points) for all models. We consider a prediction horizon of 5, 15, 30, 45 and 60 minutes, respectively, into the future (i.e. 1, 3, 6, 9 and 12 values).

We considered the following models in our evaluation:

- Naïve Baseline. The prediction is the last value observed, regardless of the prediction horizon.
- DNN. It is a one layer dense neural network. The first layer has $N \cdot T'$ nodes, output layer has $N \cdot T$, where N is the number of vertices $|\mathcal{V}|$ in the road graph, T' is the history size, and T is the number of steps to predict.
- LSTM, GRU. Vanilla three layers stacked LSTM and GRU, with a hidden state of 128 nodes, dropout of 0.2, as implemented in PyTorch.
- TGC-LSTM. We use most of the same parameters as in [5]. We use $K = 3$, i.e. 3 steps behind. For the FFR matrix we use $\hat{A}^k$ as a proxy.
- GANNSTER-LSTM, GANNSTER-GRU, our proposed models, in LSTM and GRU flavours. Implemented using two layers stacked, with a hidden dimension of 128, and dropout of 0.2, and using the adjacency matrix for the GANNSTER vectors. We use $K = 3$, that is, 3 steps behind, as a sufficient history intake.

All models have been implemented using PyTorch [23]. Source code available at https://github.com/csalort/GANNSTER. The metrics used for forecast comparison are Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) for n samples, given ground truth y and prediction $\hat{y}$:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y - \hat{y}| \tag{6}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y - \hat{y}}{y}\right| \cdot 100 \tag{7}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y - \hat{y})^2} \tag{8}$$

All experiments were conducted on a KunLun Mission Critical Server with 768 cores (equipped with Intel(R) Xeon(R) CPU E7-8890 v4 @ 2.20GHz) and 12TB of RAM.

5 Results and Discussion

Table 1 presents the results of the forecast experiments on the MUSTARD-S dataset, given as error between ground truth and prediction. First, we can observe that the forecasting accuracy generally drops as the prediction horizon is widened. We can also see that all the models have relatively poor performance, especially regarding to MAPE. This confirms our hypothesis that the high variability of the measurements create a challenging forecast environment.

Table 1: MUSTARD-S results. Best performance highlighted in bold.

		GANNSTER		TGC-	LSTM	GRU	DNN	Naïve
		GRU	LSTM	LSTM				Baseline
5 min	MAE	**2.346**	2.373	3.340	2.628	2.624	2.390	2.551
	MAPE	42.769	42.525	69.097	**41.460**	42.165	43.374	50.000
	RMSE	**4.055**	4.144	7.763	5.651	5.629	4.259	4.884
15 min	MAE	2.444	**2.407**	3.340	2.658	2.659	2.458	2.587
	MAPE	43.377	43.010	69.077	**41.781**	42.598	44.493	50.377
	RMSE	4.294	**4.246**	7.763	5.688	5.685	4.398	4.957
30 min	MAE	2.476	**2.443**	3.340	2.703	2.691	2.529	2.687
	MAPE	43.979	43.394	69.068	**42.485**	43.605	45.848	51.404
	RMSE	4.366	**4.345**	7.763	5.743	5.736	4.535	5.249
45 min	MAE	**2.567**	2.598	3.341	2.738	2.738	2.605	2.806
	MAPE	45.238	45.282	69.077	**43.008**	44.467	47.354	52.645
	RMSE	**4.634**	4.817	7.765	5.795	5.792	4.690	5.618
60 min	MAE	**2.619**	2.709	3.341	2.799	2.814	2.676	2.931
	MAPE	46.192	47.126	69.147	**44.753**	46.210	48.786	54.053
	RMSE	**4.825**	5.116	7.764	5.937	5.899	4.852	6.017

The Naïve baseline is one of the models performing worst. This type of model is unable to adapt to the variability of the measurements. Similar results can

be observed for the DNN. While both models yield low MAPE results, they perform above average in the remaining metrics. This may be caused by an overfit during hours of low traffic, generating an overly low prediction model. On the contrary, RNN performs much better in MAPE. RNN can better adapt to traffic peaks and have some of the best scores in MAPE. Interestingly, the LSTM performs best in terms of MAPE. This is because MAPE results in a disproportionately high error in case of relatively small (true and predicted) traffic volumes. LSTM is similar to real values when there is not much traffic, but when the number of cars increases it stops performing so well. TGC-LSTM, the state-of-the-art model, performs quite poorly. Our hypothesis is that using $\hat{A}$ as a proxy for the FFR matrix hurts the model. It performs worse than all the baselines, thus making it unsuitable for the properties of the dataset. Our models, GANNSTERGRU and GANNSTERLSTM, are the best performers in two out of the three metrics, and rank second in the remaining. Moreover, the accuracy improvements of GANNSTERGRU and GANNSTERLSTM over GRU and LSTM, respectively, can be attributed to the incorporation of the spatio-temporal information into the model. GANNSTER-models improve the forecast with respect to both baselines and state-of-the-art, therefore being the best suited model for the problem at hand.

6 Conclusion and Future Work

In this paper, we present GANNSTER, a graph-based RNN model designed to forecast road traffic. Our experimental evaluation compares GANNSTER with state-of-the-art methods and baselines on a real-world dataset, which has been made public. We demonstrate through a performance analysis that GANNSTER outperforms the state-of-the-art in traffic flow forecast.

Our future lines of research include the possibility to use GANNSTER on "hidden traffic metrics", by further exploiting the intrinsic spatio-temporal mechanisms at its core. A different line of research is to incorporate more long-term traffic dynamics into GANNSTER to enable a prediction horizon of days, enabling additional use-cases, such as improved city planning. Finally, we aim to extend the traffic datasets we used for evaluation, covering a larger area over a longer time, thus exploring the dynamics and robustness of the system at scale.

Acknowledgements

We want to thank our colleagues from Huawei for the insight and expertise provided during the development and writing of the paper. We would also like to thank the multiple reviewers of the paper for their useful critiques, which allowed us to write a better paper and to improve future lines of research. Jan Baumbach is grateful for the funding from H2020 project FeatureCloud, under grant agreement No 826078.

References

1. Ahmed, M.S., Cook, A.R.: Analysis of freeway traffic time-series data by using Box-Jenkins techniques. No. 722 (1979)
2. Battaglia, P., Pascanu, R., Lai, M., Rezende, D.J., et al.: Interaction networks for learning about objects, relations and physics. In: Advances in neural information processing systems. pp. 4502–4510 (2016)
3. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203 (2013)
4. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
5. Cui, Z., Henrickson, K., Ke, R., Wang, Y.: Traffic graph convolutional recurrent neural network: A deep learning framework for network-scale traffic learning and forecasting. IEEE Transactions on Intelligent Transportation Systems (2019)
6. Defferrard, M., Bresson, X., Vandergheynst, P.: Convolutional neural networks on graphs with fast localized spectral filtering. In: Advances in neural information processing systems. pp. 3844–3852 (2016)
7. Fout, A., Byrd, J., Shariat, B., Ben-Hur, A.: Protein interface prediction using graph convolutional networks. In: Advances in neural information processing systems. pp. 6530–6539 (2017)
8. Guo, S., Lin, Y., Feng, N., Song, C., Wan, H.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 922–929 (2019)
9. Hamed, M.M., Al-Masaeid, H.R., Said, Z.M.B.: Short-term prediction of traffic volume in urban arterials. Journal of Transportation Engineering **121**(3), 249–254 (1995)
10. Hamilton, W., Ying, Z., Leskovec, J.: Inductive representation learning on large graphs. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems 30, pp. 1024–1034. Curran Associates, Inc. (2017), `http://papers.nips.cc/paper/6703-inductive-representation-learning-on-large-graphs.pdf`
11. Henaff, M., Bruna, J., LeCun, Y.: Deep convolutional networks on graph-structured data. arXiv preprint arXiv:1506.05163 (2015)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Jia, Y., Wu, J., Du, Y.: Traffic speed prediction using deep learning method. In: 2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC). pp. 1217–1222. IEEE (2016)
14. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. arXiv preprint arXiv:1609.02907 (2016)
15. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436–444 (2015)
16. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural computation **1**(4), 541–551 (1989)
17. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. arXiv preprint arXiv:1707.01926 (2017)
18. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: a deep learning approach. IEEE Transactions on Intelligent Transportation Systems **16**(2), 865–873 (2014)

19. Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors **17**(4), 818 (2017)
20. Ma, X., Tao, Z., Wang, Y., Yu, H., Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transportation Research Part C: Emerging Technologies **54**, 187–197 (2015)
21. Niepert, M., Ahmed, M., Kutzkov, K.: Learning convolutional neural networks for graphs. In: International conference on machine learning. pp. 2014–2023 (2016)
22. Park, D., Rilett, L.R.: Forecasting freeway link travel times with a multilayer feedforward neural network. Computer-Aided Civil and Infrastructure Engineering **14**(5), 357–367 (1999)
23. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8024–8035 (2019)
24. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
25. Smola, A.J., Schölkopf, B.: A tutorial on support vector regression. Statistics and computing **14**(3), 199–222 (2004)
26. Van Lint, J., Hoogendoorn, S., van Zuylen, H.J.: Freeway travel time prediction with state-space neural networks: modeling state-space dynamics with recurrent neural networks. Transportation Research Record **1811**(1), 30–39 (2002)
27. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y.: Graph attention networks. arXiv preprint arXiv:1710.10903 (2017)
28. Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Yu, P.S.: A comprehensive survey on graph neural networks. CoRR **abs/1901.00596** (2019), `http://arxiv.org/abs/1901.00596`
29. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting. CoRR **abs/1709.04875** (2017), `http://arxiv.org/abs/1709.04875`
30. Yu, H., Wu, Z., Wang, S., Wang, Y., Ma, X.: Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks. Sensors **17**(7), 1501 (2017)
31. Yu, R., Li, Y., Shahabi, C., Demiryurek, U., Liu, Y.: Deep learning: A generic approach for extreme condition traffic forecasting. In: Proceedings of the 2017 SIAM international Conference on Data Mining. pp. 777–785. SIAM (2017)
32. Zhang, J., Zheng, Y., Qi, D.: Deep spatio-temporal residual networks for citywide crowd flows prediction. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
33. Zhou, J., Cui, G., Zhang, Z., Yang, C., Liu, Z., Sun, M.: Graph neural networks: A review of methods and applications. CoRR **abs/1812.08434** (2018), `http://arxiv.org/abs/1812.08434`