2018 5th International Conference on Information Science and Control Engineering

Graph Attention LSTM Network: A New Model for Traffic Flow Forecasting

WU Tianlong, CHEN Feng
Institute of Software Chinese Academy of Sciences
Beijing, China
Guiyang Academy Of Information Technology
Guiyang, China
E-mail:tianlong2015@iscas.ac.cn, chenfeng@iscas.ac.cn

WAN Yun
University of Houston-Victoria
Victoria, America
Email: wany@uhv.edu

***Abstract*—For the road networks containing multiple intersections and links, the traffic flow forecasting is essentially a time series forecasting problem on graphs. The task is challenging due to (1) complex spatiotemporal dependence among traffic flows of the whole road network and (2) sharp nonlinearity and dynamic nature under different conditions. In this paper, by extending the LSTM to have graph attention structure in both the input-to-state and state-to-state transitions, we propose the Graph Attention LSTM Network (GAT-LSTM) and use it to build an end-to-end trainable encoder-forecaster model to solve the multi-link traffic flow forecasting problem. Experiment results show that our GAT-LSTM network could capture spatiotemporal correlations better and has achieved improvement of 15% - 16% over state-of-the-art baseline.**

***Keywords*-Traffic flow forecasting; Graph Neural Network; Sequence to sequence modeling; Attention mechanism;**

I. INTRODUCTION

Traffic flow forecasting is an important application of computational intelligence and an active research topic in Intelligent Transportation Systems (ITS). This is of great significance to congestion prediction, real-time dynamic traffic signal optimization, real-time dynamic route planning, and traffic management decision. However, accurate and real-time forecasting of traffic flow in urban road network is a huge challenge. On the one hand,there is complex spatiotemporal dependence among traffic flows of the whole road network. On the other hand, traffic flows are nonlinear and dynamic, which could be influenced by many factors, such as weather, holidays, large conference or performances, traffic accidents and so on.

Traditional single-link traffic flow forecasting usually predict one link's unidirectional traffic flow at a time, which do not take the relevance of adjacent links into account. In time series community, the most popular models are Auto-Regressive Integrated Moving Average (ARIMA) and Calman filtering [1], [2], [3]. In recent years, several multi-link traffic forecasting methods have been proposed to forecast traffic flows or speeds of the whole road network simultaneously. It requires us to consider the complex spatiotemporal dependence among traffic flows of different parts of road network. To address this challenge, we could use deep neural networks. In [4], [5], [6], the authors tried to use some neural network models to solve the problem, but spatial structure was not taken into account. In [7], [8], the authors modelled the spatial correlation with Convolutional Neural Networks (CNNs), in other word, the spatial structure is in the Euclidean space (e.g., 2D images). In [9], the authors modelled the sensor network as a undirected graph and applied ChebNet and convolutional sequence model [10] for forecasting. One limitation of the mentioned spectral based convolutions is that they generally require the graph to be undirected to calculate meaningful spectral decomposition. At present, the state-of-the-art model is Diffusion Convolutional Recurrent Neural Network (DCRNN) [11], which models the traffic flow as a diffusion process on a directed graph, but this model needs to calculate the accurate weights of edges in graphs in advance, which is hard to realize in many cases.

Actually, in addition to the city road networks, there are a lot of interesting data can be viewed as graph-structured data, such as social networks, communication networks, biological networks or brain connectomes. And even regular grid data (such as image, video etc.) are also a special form of graphs. However, there have been several attempts in the literature to extend neural networks to deal with arbitrarily structured graphs. Early works used recursive neural networks to process data represented in graph domains as directed acyclic graph [12], [13]. Then, the Graph Neural Network (GNNs) is proposed as an extension of the recursive neural network to solve a more general class of graphs [14], [15]. Recently, some of the studies have achieved great results for node classification of graph-structured data [16], [17], [18]. Now, we focus on Graph Attention Networks (GATs), the latest approach, which use masked self-attentional layers to address the shortcomings of prior methods based on graph convolutions or their approximations [19].

Inspired by previous works, in this paper, we propose the Graph Attention LSTM (GAT-LSTM) Network to solve the traffic flow forecasting problem in urban road network. We define the traffic flow forecasting as a time series forecasting problem on graphs. By stacking multiple GAT-LSTM layers, we build an end-to-end trainable encoder-forecaster model to solve multi-link traffic forecasting problem. In order to

978-1-5386-5500-9/18/$31.00 ©2018 IEEE
DOI 10.1109/ICISCE.2018.00058

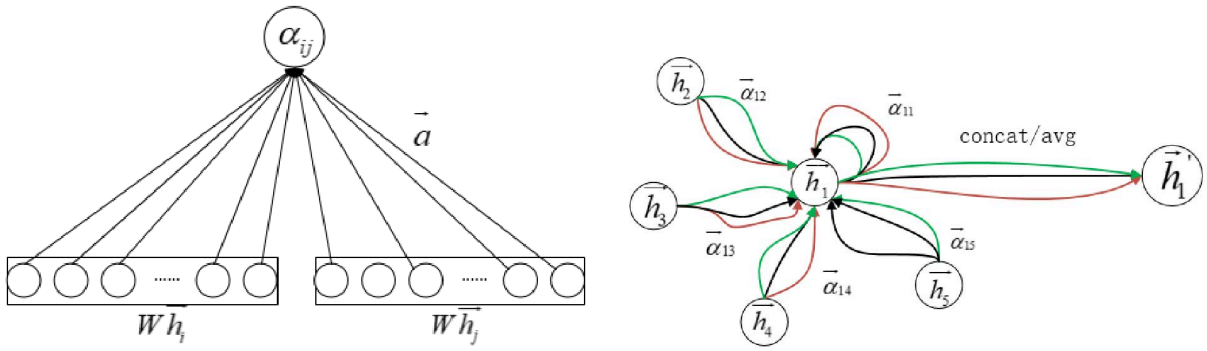


Figure 1. **Left**: The calculation of the attention coefficient between node i and node j, where $\vec{a}$ is the attention kernel. **Right**: An illustration of multi-heads ($K = 3$) by node 1 on its neighborhood. Different arrow colors denote independent attention computations. The aggregated features from each head are concatenated or averaged to obtain h'_1

validate our model, we have conducted experiments on real traffic data in Guiyang, China. The results indicated that our GAT-LSTM network captured spatiotemporal correlations better and outperformed other multi-link traffic flow forecasting methods.

II. Preliminaries

A. Definition of Multi-link Traffic Flow Forecasting

In empirical applications, we usually count the traffic flow every 5-10 minutes and forecast the next 10-30 minutes based on the historical data of the previous period. From machine learning perspective, multi-link traffic flow forecasting is a time series forecasting problem on graphs.

We assume that the road network consists of N intersections which are expressed as $insection_i$, where $i = 1, 2...N$. We can abstract this road network into a digraph, and the adjacency matrix of intersections is expressed as matrix A. If the vehicle at the $intersection_i$ can pass the $intersection_j$ in a short time (e.g. 5 minutes), there is a directed edge from the $intersection_i$ to the $intersection_j$, $A_{ij} = 1$, otherwise $A_{ij} = 0$. The maximum indegree of each intersection is four[1], so the traffic flow of the whole road network at any time can be represented by a tensor $\mathcal{X} \in \mathbb{R}^{N\times 4}$. when we count the traffic flow of the whole road network periodically, we would get a sequence of tensors $\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_t$. The multi-link traffic flow forecasting problem becomes forecasting the most likely length-K sequence in the future given the previous J observations and A.(see Equation (II-A))

$$\hat{\mathcal{X}}_{t+1}, ..., \hat{\mathcal{X}}_{t+K} = \underset{\hat{\mathcal{X}}_{t+1},...,\hat{\mathcal{X}}_{t+K}}{\arg\max} \; p(\hat{\mathcal{X}}_{t+1}, ..., \hat{\mathcal{X}}_{t+K} | \mathcal{X}_1, ..., \mathcal{X}_t, A) \tag{1}$$

B. Graph Attention Networks

Graph Attention Network (GAT) is a novel neural network architecture that operates on graph-structured data. It is an effective model with state-of-the-art results across many datasets, such as *Cora* , *Citeseer* and *protein-protein interaction* [19].

The input of Graph Attention Layer is a collection of node features, which can be expressed as $\boldsymbol{h} = \{\vec{h}_1, \vec{h}_2, ..., \vec{h}_N\}$, $h_i \in \mathbb{R}^F$, where N is the number of nodes, and F is the number of features in each node. The layer produces a new set of node features, $\boldsymbol{h'} = \{\vec{h}'_1, \vec{h}'_2, ..., \vec{h}'_N\}$, $h'_i \in \mathbb{R}^{F'}$, as its output. The transformation between input and output uses a trainable matrix shared by each node, $W \in \mathbb{R}^{F\times F'}$. In addition, the authors performed self-attention on the nodes—a shared attentional mechanism, to indicate the importance of node j's features to node i, expressed as α_{ij} which is parametrized by a attention kernel $\vec{a}^T \in \mathbb{R}^{2F'}$ (see Figure 1). The key equations are shown in (2), where $.^T$ represents transposition , $||$ is the concatenation operation and $\mathcal{N}_i = \{k|A_{ik} = 1\}$ is a collection of neighborhoods of node i in the graph.

$$\begin{aligned} \alpha_{ij} &= \frac{exp(\vec{a}^T[W\vec{h}_i \,||\, W\vec{h}_j])}{\sum_{k\in\mathcal{N}_i} exp(\vec{a}^T[W\vec{h}_i \,||\, W\vec{h}_k])} \\ \vec{h}'_i &= \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij} W\vec{h}_j) \end{aligned} \tag{2}$$

To stabilize the learning process of self-attention, K independent attention mechanisms execute the transformation of Equations (2) (see Figure 1), and then their features are concatenated or averaged, resulting in the following output feature representation:

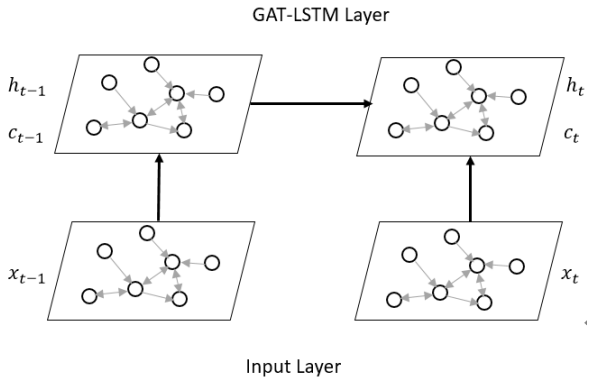
$$\begin{aligned} \vec{h}'_i &= ||_{k=1}^{K} \sigma(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j) \\ \vec{h}'_i &= \sigma(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k W^k \vec{h}_j) \end{aligned} \tag{3}$$

III. The Model

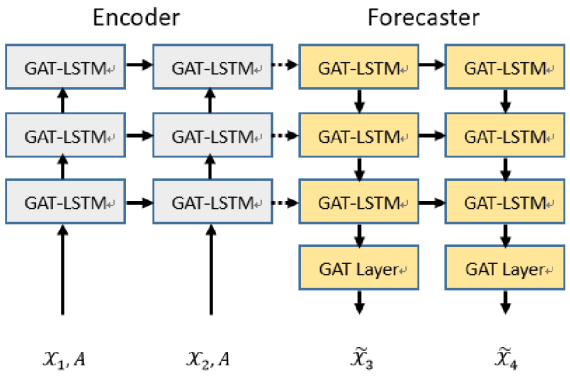
A. Graph Attention LSTM

LSTM is not suitable for handling spatiotemporal data (e.g. images or graphs), because it has to unfold the inputs to 1D vectors before processing and, as a result, all the spatial information would be lost. If we want to use a "special LSTM" for sequence modeling on graphs, we should keep the original graph unchanged in both the input-to-state and state-to-state transitions. GAT is essentially a special feature transformation of the nodes in the graph. We can

[1]If the indegree of an intersection is less than 4 (e.g. a junction of three roads), we assume the traffic flow in some directions is always 0



(a) Transformation in GAT-LSTM



(b) Encoding-forecasting structure

Figure 2. **(a)**: The transformation among input, state and output in GAT-LSTM,the arrows represent the "$\mathcal{G}$" operation. The essence of "$\mathcal{G}$" is a special feature transformation of the nodes in the graph. **(b)**: Schematic of encoding-forecasting network, in which $\mathcal{X}_1$ and $\mathcal{X}_2$ denote input sequences, A denotes adjacency matrix,$\hat{\mathcal{X}}_3$ and $\hat{\mathcal{X}}_4$ denote output of forecaster network,that are the forecasting value of $\mathcal{X}_3$ and $\mathcal{X}_4$. The dotted line between Encoder and Forecaster denote the copy of state and cell output.

define this transformation as $\mathcal{G}$: $\mathcal{G}(W, h, a, A) = h'$, where a is the attention kernel. By extending the LSTM to apply this operation in both the input-to-state and state-to-state transitions, we propose the Graph Attention LSTM Network (GAT-LSTM). All the inputs $\mathcal{X}_1, ..., \mathcal{X}_t$, cell outputs $\mathcal{C}_1, ..., \mathcal{C}_t$, hidden states $\mathcal{H}_1, ..., \mathcal{H}_t$, biases $\mathcal{B}_1, ..., \mathcal{B}_t$ and gates i_t,f_t,o_t of the GAT-LSTM have the same graph structure. The key equations of GAT-LSTM are shown in (4)

$$\begin{aligned} i_t &= \sigma(\mathcal{G}(W_{xi}, \mathcal{X}_t, a, A) + \mathcal{G}(W_{hi}, \mathcal{H}_{t-1}, a, A) + \mathcal{B}_i) \\ f_t &= \sigma(\mathcal{G}(W_{xf}, \mathcal{X}_t, a, A) + \mathcal{G}(W_{hf}, \mathcal{H}_{t-1}, a, A) + \mathcal{B}_f) \\ o_t &= \sigma(\mathcal{G}(W_{xo}, \mathcal{X}_t, a, A) + \mathcal{G}(W_{ho}, \mathcal{H}_{t-1}, a, A) + \mathcal{B}_o) \\ \mathcal{C}_t &= f_t \circ \mathcal{C}_{t-1} + i_t \circ tanh(\mathcal{G}(W_{xc}, \mathcal{X}_t, a, A) \\ &\quad + \mathcal{G}(W_{hc}, \mathcal{H}_{t-1}, a, A) + \mathcal{B}_c) \\ \mathcal{H}_t &= o_t \circ tanh(\mathcal{C}_t) \end{aligned} \tag{4}$$

On the one hand, during the input-to-state transitions, the GAT-LSTM would take into account the "graph" structural information, which is spatial information. On the other hand, during the state-to-state transitions, it would take into account temporal correlation and spatial correlation simultaneously (see Figure 2). By stacking multiple GAT-LSTM layers, the receptive field becomes larger and larger with the number of layers increasing, and we can get more abstract node features.

B. Encoding-Forecasting Structure

Similar to LSTM, GAT-LSTM can also be adopted as a building block for more complex structures. For multi-link traffic flow forecasting problem, we use the structure shown in Figure 2(a) which consists of two networks, an encoding network and a forecasting network. The reason to reverse the order of the forecasting network is that the high-level states, which have captured the global spatiotemporal representation, could guide the update of the low level states and the low-level states could further influence the forecasting [20]. Like in [20], the initial state and cell output of forecasting network is a copy of the final state and cell output of encoding network. In this way, the forecasting network give the forecasting sequence according to the road network and information encoded by encoding network (see Equations (5)). Both encoding network and forecasting network are stacked by several GAT-LSTM layers, and zeros are fed as input to the top layer in forecasting network. For our mission, we use a Graph Attention Layer to carry out the final feature transformation so that the outputs have the same structure as the input.

$$\begin{aligned} \hat{\mathcal{X}}_{t+1},...,\hat{\mathcal{X}}_{t+K} &= \underset{\hat{\mathcal{X}}_{t+1},...,\hat{\mathcal{X}}_{t+K}}{\arg\max} \; p(\hat{\mathcal{X}}_{t+1}, ..., \hat{\mathcal{X}}_{t+K} | \mathcal{X}_1, ..., \mathcal{X}_t, A) \\ &\approx \underset{\hat{\mathcal{X}}_{t+1},...,\hat{\mathcal{X}}_{t+K}}{\arg\max} \; p(\hat{\mathcal{X}}_{t+1}, ..., \hat{\mathcal{X}}_{t+K} | f_{encoding}(\mathcal{X}_1, ..., \mathcal{X}_t, A)) \\ &\approx g_{forecasting}(f_{encoding}(\mathcal{X}_1, ..., \mathcal{X}_t, A)) \end{aligned} \tag{5}$$

IV. Experiments

A. Dataset

TF-GY Dataset: Because there is no public multi-link traffic flow dataset, we build a multi-link traffic flow dataset on the basis of the "Guiyang traffic data platform" (see http://www.guiyangdata.gov.cn), and all the experiments have been carried out on this dataset. The road network used by this dataset is consisted of 112 intersections (nodes), and the raw data are from September 1st to October 31st, 2017, totaling 61 days (including working days, weekends and Chinese National Day). The training data contains 51 days and the testing data contains 10 days (a week and 3 days during Chinese National Day, 1/10, 4/10, 7/10). Because the number of vehicles is very small during 00:00-07:00 and the fluctuation is very large, we remove this part of the dataset. We count the traffic flow every 5 minutes, so each link contains 204 data points per day. In summary, there are 61*204 frames $\mathcal{X} \in \mathbb{R}^{112 \times 4}$ in our dataset $\mathcal{D}$. The road network is abstracted as a directed graph. And if the vehicle at the $intersection_i$ can pass the $intersection_j$ in 5 minutes, there is a directed edge from the $intersection_i$ to the $intersection_j$, that is $A_{ij} = 1$.

B. Experimental Settings

In order to make the result convincing, we have compared our model with results from other methods, including DCRNN, DCRNN-unweighted(same as DCRNN, we just replace the adjacency matrix in DCRNN with an unweighted adjacency matrix), LSTM, FNN. In all experiments, we obtain the forecasting of next 30 minutes, based

Table I
COMPARISON OF DIFFERENT APPROACHES.

	FNN	LSTM	DCRNN	DCRNN-UNWEIGHTED	GAT-LSTM
5 MIN	33.17	24.34	18.12	15.65	13.15
10 MIN	32.67	24.91	18.67	16.24	13.80
15 MIN	33.27	25.54	18.98	16.88	14.28
20 MIN	33.10	26.45	19.20	17.47	14.76
25 MIN	33.04	26.90	19.97	18.07	15.21
30 MIN	33.02	27.14	20.13	18.58	15.60

on the data of the previous one hour. In this case, a training sequence consists of 18 consecutive frames in dataset $\mathcal{D}$. There are 187 training sequences every day, so we have 51*187 training sequences and 10*187 testing sequences. We use the root mean squared error (RMSE) as the loss function and Adam as the optimizer. Besides, we use Z-score standardization for data preprocessing.

Table I shows the comparison of different approaches range from 5 minutes to 30 minutes ahead forecasting on TF-GY Dataset. GAT-LSTM achieves best performance for all forecasting horizons.

It is noteworthy that DCRNN-unweighted outperforms DCRNN on this dataset. The cause of this phenomenon may be that we have calculated the weights of edges in a wrong way. For traffic speed forecasting, the relationship between sensors could be accurately described by the distance between them. But, for traffic flow forecasting, the relationship between intersections is dynamic. Many factors, such as the tidal phenomenon of traffic flow, traffic congestion and weather and so on, can change the travel time between the intersections, which can also affect the relationship between the intersections.

Since GAT-LSTM and DCRNN-unweighted are much better than other models. we only visualize their forecasting outputs.

Figure 3 shows that under normal conditions, both two models could achieve good results and their results are very close.

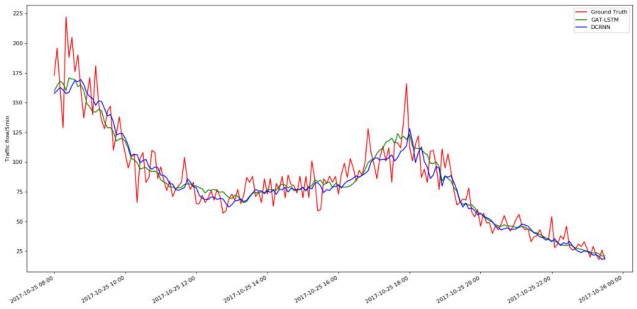


Figure 3. Under normal conditions, the ground truth and the forecasting outputs of GAT-LSTM and DCRNN-unweighted.

In Figure 4, the traffic flow suddenly decreased rapidly during 13:00 to 16:00. After communicating with the local traffic police, we are sorry to know there was a traffic accident near this intersection at about 13:50 October 30, 2017. In this extreme traffic condition, our model can still predict traffic flow relatively accurately, and after analysis, we believe that the attention mechanism in GAT-LSTM plays a decisive role. In Figure 5, we compare the attention coefficients of this intersection at 13:20 and 14:20. As we can see, the attention coefficients of this intersection is scattered at 13:20, which indicates that this intersection is associated with most of its adjacent intersections. However the attention coefficients of this intersection is concentrated at 14:20, which indicates that this intersection is almost only associated with itself. It is consistent with the actual situation, because in practice, the intersections of traffic congestion or accidents are often isolated.

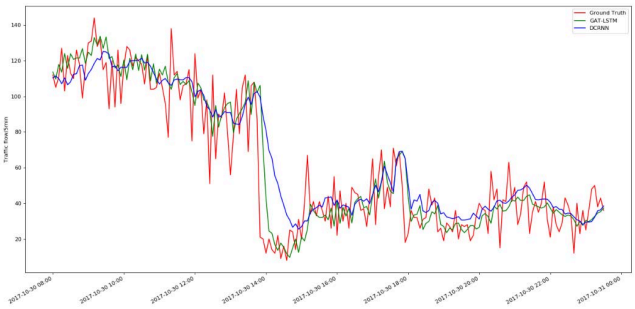
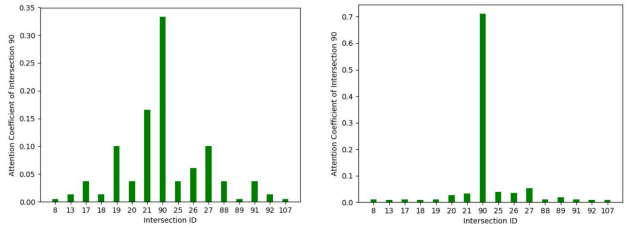


Figure 4. At about 13:50, there was a traffic accident near this intersection and the traffic flow suddenly decreased rapidly, and the GAT-LSTM can catch this change more timely.



(a) Attention coefficients of Intersection 90 at 13:20

(b) Attention coefficients of Intersection 90 at 14:20

Figure 5. The attention coefficients of the intersection in Figure **??** at 13:20 and 14:20 and there is a great difference between them. **Left**: the attention coefficients are scattered, which indicates that this intersection is associated with most of its adjacent intersections. **Right**: the attention coefficients are concentrated, which indicates that this intersection is almost only associated with itself.

V. CONCLUSION

In this paper, we propose a new extension of LSTM, called GAT-LSTM, and use it to build an encoder-forecaster model to solve the problem of multi-link traffic flow forecasting. The experiment results demonstrate that our model has achieved state-of-the-art results. Also, we can easily find that the model could be applied to other time series problem with irregular graph structure by constructing an appropriate adjacent matrix.

REFERENCES

[1] M. Lippi, M. Bertini, and P. Frasconi, "Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, no. 2, pp. 871–882, 2013.

[2] I. Okutani and Y. J. Stephanedes, "Dynamic prediction of traffic volume through kalman filtering theory," *Transportation Research Part B*, vol. 18, no. 1, pp. 1–11, 1984.

[3] B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of Transportation Engineering*, vol. 129, no. 6, pp. 664–672, 2003.

[4] Y. Gao and S. Sun, "Multi-link traffic flow forecasting using neural networks," in *International Conference on Natural Computation*, 2010, pp. 398–401.

[5] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, "Traffic flow prediction with big data: A deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[6] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, *Deep Learning: A Generic Approach for Extreme Condition Traffic Forecasting*, 2017.

[7] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, 2017.

[8] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y. Wang, "Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction," *Sensors*, vol. 17, no. 4, 2017.

[9] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional neural network: A deep learning framework for traffic forecasting," 2017.

[10] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," 2017.

[11] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," 2017.

[12] P. Frasconi, M. Gori, and A. Sperduti, "A general framework for adaptive processing of data structures," *IEEE Transactions on Neural Networks*, vol. 9, no. 5, p. 768, 1998.

[13] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Transactions on Neural Networks*, vol. 8, no. 3, pp. 714–735, 1997.

[14] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings*, 2005, pp. 729–734 vol. 2.

[15] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Transactions on Neural Networks*, vol. 20, no. 1, pp. 61–80, 2009.

[16] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," 2016.

[17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016.

[18] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," 2017.

[19] P. Velikovi, G. Cucurull, A. Casanova, A. Romero, P. Li, and Y. Bengio, "Graph attention networks," 2017.

[20] X. Shi, Z. Gao, L. Lausen, H. Wang, D. Y. Yeung, W. K. Wong, and W. C. Woo, "Deep learning for precipitation nowcasting: A benchmark and a new model," 2017.