GLGAT: Global-Local Graph Attention Network For Traffic Forecasting

1st Qi Shao
School of Mathematics
Southeast University
Nanjing, China
seushaoqi@163.com

2nd Yifan Zhang
School of Information Science and Engineering
Southeast University
Nanjing, China
213170403@seu.edu.cn

3rd Duxin Chen
School of Mathematics
Southeast University
Nanjing, China
chendx@seu.edu.cn

4th Wenwu Yu
School of Mathematics
Southeast University
Nanjing, China
wwyu@seu.edu.cn

Abstract—**Spatiotemporal prediction plays an important role in many area, especially intelligent transportation system (ITS). However, due to the complex spatiotemporal conditions, traffic flow prediction task is challenging. In this paper, we propose a novel GLGAT model to predict spatiotemporal traffic flow. GLGAT can grasp global and local spatiotemporal information respectively, and make accurate predictions of the traffic flow. We test GLGAT on PeMSD7 data set and the experiment results indicate that GLGAT could outperform the state-of-the-art baselines especially in long-term prediction.**

Index Terms—**Traffic forecasting, Graph neural networks, Attention mechanism.**

I. Introduction

Spatiotemporal flow data prediction is a hot spot in the research of prediction problems in recent years, and it has a wide range of applications in many fields, such as autonomous driving, smart grid optimization, traffic flow prediction and so on. In this paper, we have focus on traffic flow forecasting, which is one of the core research issues of intelligent transportation system). Achieving effective traffic flow forecasting will help provide pedestrians with real-time and effective information, help them to make better route selection, realize route guidance, and reduce travel time and traffic congestion.

Specifically, traffic flow is the number of vehicles passing through the observation nodes on traffic networks at each time interval. The goal of traffic flow prediction is to predict the future traffic flow for several times based on historical traffic data and physical traffic network. In recent years, many researchers have done meaningful research on the traffic flow prediction problem and put forward many effective methods [1], [2], which also provide the basis for further research.

The research on traffic flow forecasting has a long history and can be divided into two categories: statistical-based models and data-driven models. In the research of traditional statistical methods, the prediction of short-term traffic flow is often emphasized. The main statistical models related to traffic flow prediction are historical average model (HA) [3], time series model (ARIMA) [4], [5], Kalman filter model (KF) [6], [7], etc. Traditional statistical models rely on mathematical models and have good interpretability, but the model is difficult to reflect the true changes in the data. At the same time, when the predicted step size increases, the calculation amount of the traditional model will increase rapidly, which also limits the application of traditional model.

The rise of neural network algorithms [8] has opened up new ideas for traffic flow forecasting. Since neural network algorithms relys on data-driven model, and get rid of the trouble of establishing accurate mathematical models, they can adapt to more complex traffic conditions, and have great potential in the field of traffic flow prediction. In 1992, Clark et al. [9] introduced BP neural networks into traffic prediction and achieved better results than traditional models. Since then, more and more studies have paid attention to neural network methods.

Since BP neural network only focuses on predicting traffic flow under a specific condition [10], the model lacks generalization. While traffic flow prediction is a time-series forecasting problem, BP network cannot effectively learn time characteristics and has limitations. In this case, the LSTM (Long Short-Term Memory) [11] in 1997 and the GRU (Gated Recurrent Unit) [12] in 2014 provided a new solution to the traffic flow prediction problem, and also promoted the development of time series information flow prediction technology [13]. LSTM and GRU are two variants of Recurrent Neural Network (RNN) [14]. In 2016, Fu et al. introduced LSTM and GRU methods to the field of traffic flow prediction, and used the LSTM framework for traffic flow prediction, which provided a new framework for future work [15].The RNN model can effectively extract the time series information of the traffic flow, but the traffic flow prediction problem should not only consider the time series relationship of the data, but also consider the spatial information of the traffic flow network; at the same time, the RNN structure can extract the short-term dependence of the time-series data and the long-term dependence. while the extraction of time dependence needs to be improved.

In order to solve the problem of spatial information feature extraction in traffic flow prediction, graph convolution provides a new method for spatial feature extraction [16]. Since then, the graph convolution method was gradually introduced into the field of traffic flow prediction, and the prediction framework combining graph convolution and RNN has gradually become the mainstream method of traffic flow prediction models. In recent years, traffic forecasting articles have used graph convolutional networks and time series forecasting structures to form ST spatiotemporal forecasting models, and have obtained good results [1], [2]. However, those models are not good at long-term prediction, since the model cannot learn the long-term dependence of time series data.

In this work, we propose the Global-Local Graph Attention Network(GLGAT) mdoel to learn the long-term dependence for traffic forecasting. Through the self-attention mechanism [17], the GLGAT model can effectively learn the long-term relationship of data. At the same time, the GLGAT model contains rucurrrent neural network structure that can summarize the local relationship of data and ensure the short-term stability of the model. The contributions of this article are as follows:

1) GLGAT can effectively grasp the long-term dependence of data and provide a method for solving long-term prediction.

2) GLGAT uses the attention mechanism to realize the dynamic processing of graph structure data, and the model is also inductive.

3) Compared with existing results, long-term prediction results of GLGAT are significantly better than the baselines.

II. Related Work

A. Deep learning on traffic forecasting

In recent years, deep learning methods have been widely used in traffic prediction. In 2015, Lv et al. used SAE to predict the flow status of different nodes [18]. In 2017, Zhao et al. applied LSTM network to short-term traffic flow prediction and obtained good results [19]. In 2016, He et al. used CNN with residual connections for image recognition [20]. The similar idea was also adopted in traffic forecasting [21]–[23]. Some of these works also embed LSTM networks or attention mechanisms to enhance model performance. However, none of these methods can extract the spatial information of the network.

B. Deep learning on Graphs

Graph convolutional network is an important classification of graph neural network. Graph convolutional network draws on the idea of convolution in convolutional neural network, uses convolution to process the information of graph structure data, and obtains good information characteristics. Since the data of the graph structure is a non-Euclidean data structure, the convolutional network cannot be directly used for feature extraction. In 2014, Bruna et al. proposed the first-generation graph convolutional network using the Laplacian matrix of the graph [24]. However, this method requires the eigen decomposition of the Laplacian matrix, so the amount of calculation is huge. In 2016, Defferrard et al. improved the convolution kernel using Chebyshev polynomials and proposed chebNet [25], which greatly reduced the computational complexity, and then kipf Others further modified chebnet and proposed a common graph convolution structure [16].

In 2017, Hamilton et al. proposed the GraphSage algorithm to realize transferable learning of graph structure data [26]. GraphSage uses the neighbor information of the node to update the node information by aggregating the information of the node itself and its neighbors through the aggregation layer. Since the aggregation layer does not depend on the number of node neighbors, it can satisfy graph data of different structures and realize the inductive task. In 2018, Veličković et al. proposed the graph attention network (GAT) [27], which introduced the aggregation layer into the attention mechanism. By calculating the scores of each neighbor node and the central node, different weights were obtained to update node information. GAT can realize the inductive task.

With the development of GCN, many works introduced GCN into traffic forecasting area and proposed spatiotemporal traffic forecasting model. Di et al. combined LSTM and MGCN to use multi-graph GCN to extract information and LSTM to process time series features to complete the shared bicycle traffic prediction [28]. Xu et al. combined RNN and GCN, and proposed ST-MGCN to realize online car-hailing demand prediction [29]. Guo et al. proposed ASTGCN, which uses three components to model the three modes of time series: the near term, daily cycle, and weekly cycle. Each component uses K-order Chebyshev graph convolution to capture the spatial relationship, using one-dimensional Convolution captures the time relationship. The performance is better than STGCN In the PEMS data set [30]. Zhang et al. used the GAT algorithm combined with GRU to obtain the model GaAN, which has good mobility and obtained significant results [31]. In this case, we also apply spatiotemporal model for traffic prediction.

III. Methodology

In this section, the proposed GLGAT model will be introduced. Specifically, the overall architectures of the proposed model will be introduced and the details of the GLGAT model will also be elaborated.

A. Problem definition

The network structure of a traffic flow can be defined as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. There are traffic flow time series $X_1, X_2, ..., X_T$ of each node. The goal of traffic flow forecasting is to use the traffic flow network G and time series $X_1, X_2, ..., X_T$ to predict the traffic situation at the next H time:

$$\widehat{\mathrm{X}}_{\mathrm{T}+1}, \widehat{\mathrm{X}}_{\mathrm{T}+2}, \ldots, \widehat{\mathrm{X}}_{\mathrm{T}+\mathrm{H}} = f\left(\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{\mathrm{T}}, \mathrm{G}\right) \tag{1}$$

Where $\widehat{\mathrm{X}}_{\mathrm{T}+1}, \widehat{\mathrm{X}}_{\mathrm{T}+2}, \ldots, \widehat{\mathrm{X}}_{\mathrm{T}+\mathrm{H}}$ is the prediction value of the time step H+1 to the time step T+H;$\mathrm{X}_1, \mathrm{X}_2, \ldots, \mathrm{X}_{\mathrm{T}}$ is the traffic flow value of time step 1 to time step T; $\mathcal{G}$ is the graph of the flow data. The goal of the problem is to learn the function f, which makes prediction value $\widehat{\mathrm{X}}_{\mathrm{T}+1}, \widehat{\mathrm{X}}_{\mathrm{T}+2}, \ldots, \widehat{\mathrm{X}}_{\mathrm{T}+\mathrm{H}}$ as close to the true value as possible.

B. Model Sturcture Overview

The overall structure of the proposed GLGAT model is shown in Fig. 1 , which consists of the stacked spatiotemporal blocks (ST-Blocks), a data dimension promotion layer, and a prediction layer. More specifically, the data dimension layer

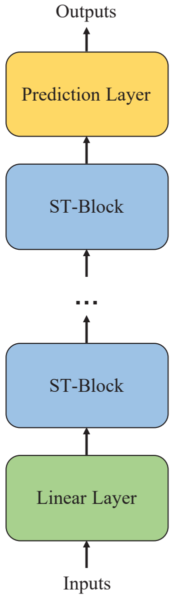


Fig. 1. GLGAT Structure

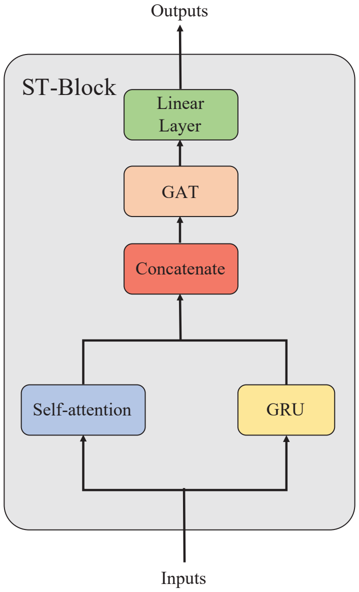


Fig. 2. ST-Block Sturcture

contains a linear transformation, which can upgrade low-dimensional data to high-dimensional data to facilitate feature extraction. Each ST-Block contains a temporal feature extraction part and a spatial feature extraction part which can extract temporal information and spatial information respectively. In GLGAT model, the number of ST-Blocks is set as 2, and each ST-block will be connected with ResNet [20]. The prediction layer consists of a sequence of linear layers and activation functions, which can reduce the data dimension to the forecast dimension.

C. Sturcture of Spatiotemporal Module

The ST-blocks contain two parts: the temporal part and the spatial part. As shown in Fig. 2, the temporal parts consists of a self-attention part and a GRU part.

The self-attention part can grasp global time information with the self-attention mechanism. Supposed the input data is $\mathcal{X} \in R^{N*D*T}$, which means the data consists of N nodes, D features and T steps. For each time t, the data $\mathcal{X}_t$ can be upgraded by self-attention mechanism:

$$\begin{gathered} Q_i = \mathcal{X}_i * W_Q \\ K_i = \mathcal{X}_i * W_K \\ V_i = \mathcal{X}_i * W_V \\ \mathcal{X}_t^{'} = Attention(Q_i, K_i, V_i) = Softmax\left(\frac{Q_i * K^T}{\sqrt{d_k}}\right) V_i \end{gathered} \tag{2}$$

Where d_k is the dimension of K. From the equation (2), it can be seen each $\mathcal{X}_t$ will calculate its own Q, K, V vector and each $\mathcal{X}_t$ will use its Q vector to calculate attention scores with all K vectors. In this case, the self-attention mechanism can grasp global time information.

The GRU part is a traditional RNN structure which can be described as :

$$\begin{aligned} z_t &= \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \\ r_t &= \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \\ \tilde{h}_t &= \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right) \\ h_t &= (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \end{aligned} \tag{3}$$

Where h_t is the hidden layer and the $h_0 = \mathcal{X}_0$. As the equation (3) shown, the GRU structure can learn the sequence information of the time series data, which means the GRU model can grasp local time information.In temporal part, the input $\mathcal{X}$ passes the self-attention part and GRU part respectively, and the last time step output of each part will be concatenated as the temporal part output.

After the temporal part, the temporal output $\mathcal{Y} \in R^{N*F}$ will be obtained, where N is the nodes number and F is the nodes feature number. Then the output Y will be used as the input of GAT part for spatial information extraction. GAT model is usually used to extract spatial information of network structure data. Supposed the input $\mathcal{Y} = [\mathcal{Y}_1, \mathcal{Y}_2, \ldots, \mathcal{Y}_N]$, and the output of GAT model can be described as $\mathcal{Y}' = [\mathcal{Y}_1', \mathcal{Y}_2', \ldots, \mathcal{Y}_N'] \in R^{N*F'}$. In other words, GAT model can generate new node representation of the graph data. More specifically, GAT can be calculated with:

$$\begin{gathered} e_{ij} = a\left(W\vec{\mathcal{Y}}_i, W\vec{\mathcal{Y}}_j\right) \\ \alpha_{ij} = \operatorname{softmax}_j\left(e_{ij}\right) = \frac{\exp(e_{ij})}{\sum_{k \in N_i} \exp(e_{ik})} \\ \alpha_{ij} = \frac{\exp\left(\text{ LeakyReLU }\left(\vec{a}^T\left[W\vec{\mathcal{Y}}_i \| W\vec{\mathcal{Y}}_j\right]\right)\right)}{\sum_{k \in \mathcal{N}_i} \exp\left(\text{ LeakyReLU }\left(\vec{a}^T\left[W\vec{\mathcal{Y}}_i \| W\vec{\mathcal{Y}}_k\right]\right)\right)} \\ \vec{\mathcal{Y}}_i' = \sigma\left(\sum_{j \in N_i} \alpha_{ij} W\vec{\mathcal{Y}}_j\right) \end{gathered} \tag{4}$$

As equation(4) shown, each node of the graph will be calculated with its neighbor nodes to get the attention coefficient α_{ij} in GAT model. In this case, the GAT output of each node will only be related to neighbor nodes which realizes the induction of the model.

After the GAT part, the spatial information was extracted and the ST-Block finished the spatiotemporal information extraction. Since the data dimension was decreased in temporal block(from R^{N*T*D} to R^{N*F}), the linear transformation will be used to promote data dimension. Then the ST-Block output $\mathcal{Z} \in R^{N*T*F'}$.

IV. Experiments

A. Datasets

In this paper, we use PeMSD7(M) data set [1] to test the performance of GLGAT algorithm. The PeMSD7(M) data set is a part of the PeMS data set, which contains traffic flow data over 44 days for 228 traffic stations in District 7 of California. In PeMSD7, the speed of traffic flow is sampled at the interval of 5 minutes. In the experiment, we used 34 of the 44 days for training, 5 days for validation, and the rest 5 days for testing.

B. Experimental Settings

The model is trained on a NVIDIA Tesla P100. The loss function is the mean squared error (MSE) between the predicted traffic flow speed and the real data and we use Adam optimizer to train GLGAT. The learning rate is set as 0.001 and the batch size is set as 32. Early stopping [32] is also applied in the training to prevent the model from overfitting. GLGAT will automatically stop training when the MSE of validation set does not decrease for 5 epochs.

C. Experimental Results

We input the data of an hour (12 sample points) into GLGAT, and the model outputs the prediction of the traffic flow for the next hour. In order to illustrate the prediction effect of the model, we mainly observed the predictions of the next 15, 30 and 60 minutes. The performance of the model is evaluated by three metrics: mean absolute error (MAE), mean absolute percentage error (MAPE) and root mean square error (RMSE), which are defined as follows:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i| \tag{5}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}|\frac{\hat{y}_i - y_i}{y_i}| \tag{6}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2} \tag{7}$$

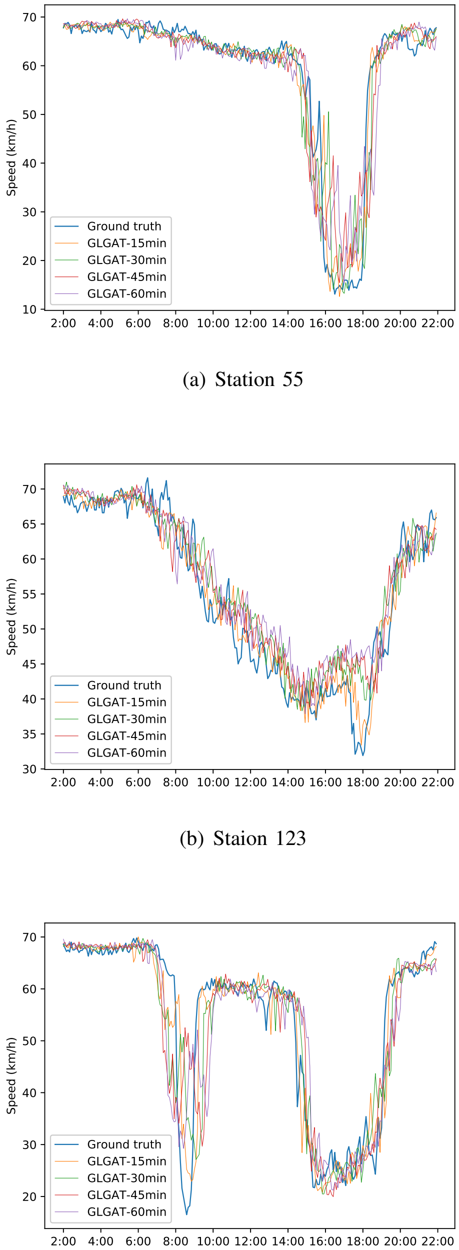
TABLE I demonstrates the results of our GLGAT model and compares it with other widely used traffic flow prediction models on PeMSD7(M) data set.

TABLE I
PERFORMANCE OF DIFFERENT MODELS ON PEMSD7(M).

Model	PeMSD7(M) (15/30/60 min)		
	MAE	MAPE (%)	RMSE
HA	4.01	10.61	7.20
LSVR	2.49/3.46/4.94	5.91/8.42/12.41	4.55/6.44/9.08
FNN	2.53/3.73/5.28	6.05/9.48/13.73	4.46/6.46/8.75
FC-LSTM	3.57/3.92/4.16	8.60/9.55/10.10	6.20/7.03/7.51
STGCN	**2.24/3.02**/4.01	**5.20/7.27**/9.77	**4.07**/5.70/7.55
GLGAT(ours)	2.49/3.21/**3.90**	5.87/8.00/**9.73**	4.20/**5.57/6.73**

It can be found that neural networks based algorithms (e.g, FNN and FC-LSTM) usually have better performance than traditional algorithms for time series prediction (e.g, HA and LSVR [33]). However, algorithms like FNN [34] and FC-LSTM [35] only focus on each traffic station, which means the connections between stations and the global traffic network are ignored. As a result of using the traffic network and graph neural networks, STGCN and DCRNN make great improvement to the prediction.

Our GLGAT model outperforms these baselines in long-term prediction by considering the traffic flow from local and global perspective, respectively. Specifically, in the prediction of 60 minutes, our GLGAT model surpasses STGCN by 10.8% in RMSE indicator.



(a) Station 55

(b) Staion 123

(c) Station 125

Fig. 3. Prediction Results of 3 Stations

In order to show the effect of the prediction of GLGAT in more detail, the comparison of real traffic flow speed and predicted speed is shown in Fig.3, which shows the real and predicted traffic flow of 3 stations from 2 am to 10 pm on a certain day.

V. Conclusion and Future Work

In this paper, we have proposed an original and novel deep leaning model GLGAT for spatiotemporal traffic forecasting.

Our paper has three contributions: (1) GLGAT model uses self-attention mechanism and GRU structure to grasp global and local time series information respectively, and GAT is also applied to extract spatial information. (2) The self-attention mechanism is introduced into feature extraction of time series data information. (3) The residual network is used to enhance the ability of the model to extract the node information.

In the experiments of the PeMSD7(M) data set, our model has surpassed many novel baselines, such as STGCN [1] and DCGRU [2]. Especially in terms of long-term prediction, our model has shown significant superiority.

We will improve the model in future work, especially improve the accuracy of short-term prediction and ensure the accuracy of long-term prediction. To achieve this goal, the structure of the ST-Block should be modified. Also, we can add time or date label to the input data to obtain a more accurate result.

References

[1] B. Yu, H. Yin, and Z. Zhu, "Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting," *arXiv preprint arXiv:1709.04875*, 2017.

[2] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *arXiv preprint arXiv:1707.01926*, 2017.

[3] Y. J. S. EDES, P. G. Michalopoulos, and R. A. Plum, "Improved estimation of traffic flow for real-time control," *and Characteristics*, vol. 7, no. 9, p. 28, 1980.

[4] M. S. Ahmed and A. R. Cook, *Analysis of freeway traffic time-series data by using Box-Jenkins techniques*. No. 722, 1979.

[5] G. E. Box and D. A. Pierce, "Distribution of residual autocorrelations in autoregressive-integrated moving average time series models," *Journal of the American statistical Association*, vol. 65, no. 332, pp. 1509–1526, 1970.

[6] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

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

[8] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*, vol. 1. MIT press Cambridge, 2016.

[9] S. D. Clark, M. S. Dougherty, and H. R. Kirby, "The use of neural networks and time series models for short term traffic forecasting: a comparative study," in *TRANSPORTATION PLANNING METHODS. PROCEEDINGS OF SEMINAR D HELD AT THE PTRC EUROPEAN TRANSPORT, HIGHWAYS AND PLANNING 21ST SUMMER ANNUAL MEETING (SEPTEMBER 13-17, 1993), UMIST. VOLUME P363*, 1993.

[10] S.-M. Chin, H.-L. Hwang, and S.-P. Miaou, "Transportation demand forecasting with a computer-simulated neural network model," in *Proceedings of the International Conference on Artificial Intelligence Applications in Transportation Engineering, San Buenaventura, CA, Institute of Transportation Studies, University of California, Irvine*, 1992.

[11] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[12] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[13] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[14] M. Boden, "A guide to recurrent neural networks and backpropagation," *the Dallas project*, 2002.

[15] R. Fu, Z. Zhang, and L. Li, "Using lstm and gru neural network methods for traffic flow prediction," in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, pp. 324–328, IEEE, 2016.

[16] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[17] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[18] 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, 2014.

[19] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[21] X. Zhou, Y. Shen, Y. Zhu, and L. Huang, "Predicting multi-step citywide passenger demands using attention-based neural networks," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 736–744, 2018.

[22] H. Yao, X. Tang, H. Wei, G. Zheng, Y. Yu, and Z. Li, "Modeling spatial-temporal dynamics for traffic prediction," *arXiv preprint arXiv:1803.01254*, 2018.

[23] H. Yao, F. Wu, J. Ke, X. Tang, Y. Jia, S. Lu, P. Gong, J. Ye, and Z. Li, "Deep multi-view spatial-temporal network for taxi demand prediction," *arXiv preprint arXiv:1802.08714*, 2018.

[24] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203*, 2013.

[25] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in neural information processing systems*, pp. 3844–3852, 2016.

[26] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Advances in neural information processing systems*, pp. 1024–1034, 2017.

[27] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[28] D. Chai, L. Wang, and Q. Yang, "Bike flow prediction with multi-graph convolutional networks," in *Proceedings of the 26th ACM SIGSPATIAL international conference on advances in geographic information systems*, pp. 397–400, 2018.

[29] X. Geng, Y. Li, L. Wang, L. Zhang, Q. Yang, J. Ye, and Y. Liu, "Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, pp. 3656–3663, 2019.

[30] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "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.

[31] J. Zhang, X. Shi, J. Xie, H. Ma, I. King, and D.-Y. Yeung, "Gaan: Gated attention networks for learning on large and spatiotemporal graphs," *arXiv preprint arXiv:1803.07294*, 2018.

[32] L. Prechelt, "Early stopping-but when?," in *Neural Networks: Tricks of the trade*, pp. 55–69, Springer, 1998.

[33] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[34] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[35] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, pp. 3104–3112, 2014.