10th International Conference on Information Science and Technology
Bath, United Kingdom, September 9-15, 2020

# Traffic Flow Prediction Using Graph Convolution Neural Networks

Anton Agafonov
*Geoinformatics and Information Security Department*
*Samara National Research University*
Samara, Russia
ant.agafonov@gmail.com

***Abstract*—Traffic flow prediction using spatial-temporal network data remains one of the most important problems in intelligent transportation systems. Timely and accurate traffic prediction is necessary to provide valuable information for different urban planning, traffic control, and guidance tasks. The complexity of the problem is explained by the fact that traffic flows have high nonlinearity and complex spatial-temporal correlations. The development of mathematical models, and, in particular, the deep learning models, allows to use convolutional neural networks to solve traffic prediction problems. In this article, we analyze the architecture of the graph convolution network for traffic flow prediction. The considered graph convolution network takes into account daily and weekly patterns of traffic flow distributions. Experimental studies of the graph neural network model were carried out on the transportation network of Samara city. Experiments show that the considered model outperforms other baseline forecasting algorithms.**

***Keywords*—traffic prediction, graph neural network, convolutional neural network, intelligent transportation system**

## I. Introduction

Intelligent transportation systems (ITS) have emerged as an efficient way to improve the quality of transportation systems [1]. Nowadays, the development of ITS and its components remains one of the most popular problems in transportation planning and management from both a research and a practical point of view. ITS can be used both for transport management (for example, by regulating traffic lights, public transport schedule planning, etc.) and for providing various services to road users (including an arrival time prediction of public transport, solving the navigation problem, etc. [2]). Accurate and timely traffic flow prediction is necessary for various practical in for traffic management and urban planning. This problem can be considered as one of the most challenging components of ITS.

In [3], the authors review the short-term traffic forecasting problem, summarize main challenges, and classifies existing approaches according to various criteria, including the used data collection technologies, the predicted characteristics of traffic flows, the used models and algorithms, etc. In [4], the authors give an overview of the recent advances and new challenges in road traffic forecasting, describe the latest technical achievements and unsolved problems.

The work was partially supported by RFBR research projects nos. 18-07-00605 A, 18-29-03135-mk.

There are different traffic prediction models and algorithms classifications [3]–[5], but roughly existing models can be classified into two categories: statistical methods and machine learning methods. In classical statistical methods, it is usually assumed that transportation systems are quite simple, or use a relatively small size of traffic datasets to make predictions. Statistical methods include time series ARIMA models [6], [7], vector autoregression models [8], etc. However, the ability of statistical models to process spatio-temporal data of high dimension is quite limited.

Currently, the rapid development of high-performance computing systems, including parallel computing platforms with graphics processing units, as well as the increase in the volume of traffic data leads to the development and extensive use of machine learning methods for different practical tasks, including the traffic prediction problem. In many studies, it was shown, that machine learning methods outperform statistical methods because of their ability to model complex nonlinear relationships. In [9], [10], the support vector regression (SVR) was used to solve the short-term traffic flow prediction problem. The k nearest neighbors (kNN) method implemented as a part of the system for massive data processing with the Big Data concept was studied in [11]–[13]. However, with the increase in the dataset size, the efficiency of kNN decreases due to the high computational cost. In [14], [15], the authors investigated fully connected neural networks.

In the last decade, deep neural networks, including networks with long short-term memory (LSTM) [16], [17], convolutional neural networks [18], [19], have been widely and successfully applied to various tasks in different scientific fields. Traditional convolutional neural networks can efficiently retrieve local patterns, but they have a limitation since the input data structure must be standard 2D or 3D regular grid [20].

Convolution on graphs summarizes traditional convolution and can be used to process data on graph structures. Graph neural networks use the graph adjacency matrix to describe the graph structure [21]. Generalized graph neural networks based on the use of the graph Laplacian operator were proposed in [22]. However, it is expensive to directly perform the eigenvalue decomposition on the Laplacian matrix, therefore, Chebyshev polynomials were adopted to solve this problem approximately in [23]. For classification problems, graph networks were used in [24]. In [25], graph neural networks were used to predict traffic flows, but the spatio-temporal correlations in the data were not taken into account. In [26], the spatial-temporal attention mechanism was proposed to capture the dynamic spatial-temporal correlations in traffic data. In [27], the authors incorporate a graph convolutional neural network with long short-term memory layers. However, it is worth noting that in the considered works, data from a small number of detectors were used as the data source. In this paper, we consider the road network of the Samara city, Russia; the average travel time of road segments is used as the traffic dataset.

In this paper, we investigate the effectiveness of using graph convolutional networks for short-term traffic flow prediction. We analyze the architecture of the neural network that uses spatial-temporal correlations in traffic data.

The paper is organized as follows. In section II, the main notations and the problem statement is given. Next, we describe the architecture of the neural network. In section IV, experimental results are provided. Finally, we conclude the work and present possible directions for future research.

## II. Problem Statement

A road network is considered as a directed graph $G = (V, E, \mathbf{A})$, with nodes $V, N_V = |V|$ represent the road links, edges $E, N_E = |E|$ denote road intersection, $\mathbf{A} \in \mathbb{R}^{N \times N}$, denotes the adjacency matrix of graph $G$.

We assume that each node of graph $G$ is described by a feature vector $\mathbf{x}_t^i \in \mathbb{R}^F$ that presents the traffic flow at the vertex $i\ inV$ at a given moment time $t$ (time interval). $x_t^{j,i}\ in\ mathbbR$ determines the value of the $j$ -th element of the feature vector of the vertex $i\ inV$ at time $t$. The following traffic flow parameters can be used as elements of the feature vector:

- average speed,
- density,
- flow.

In this work, as a predicted traffic flow characteristic for the experimental study we use the average traffic speed.

Denotes the set of feature vectors for all the nodes at time $t$ as

$$\mathbf{X}_t = \left(\mathbf{x}_t^1, \mathbf{x}_t^2, \cdots, \mathbf{x}_t^N\right) \in \mathbb{R}^{N \times F}. \tag{1}$$

The value of all feature vectors for all the nodes over $\tau$ time intervals denotes as

$$\boldsymbol{\chi} = (\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_\tau)^T \in \mathbb{R}^{N \times F \times \tau}. \tag{2}$$

Let $y_t^i \in \mathbb{R}$ be the required (predicted) value of the feature vector at the future timestamp $t$ for node $i \in V$, $\mathbf{y}^i = \left(y_{\tau+1}^i, y_{\tau+2}^i, \cdots, y_{\tau+T}^i\right) \in \mathbb{R}^T$ be the required values of the feature vectors for node $i$ over $T$ time intervals, $\mathbf{Y} = \left(\mathbf{y}^1, \mathbf{y}^2, \cdots, \mathbf{y}^N\right)^T \in \mathbb{R}^{N \times T}$ be the required values of the feature vectors for all nodes of graph $G$ over $T$ time intervals.

Given the introduced notation, the short-term traffic flow forecasting problem can be formulated as follows:

*Given a graph $G(V, E, \boldsymbol{A})$ and sequence $\boldsymbol{\chi}$ of observed historical traffic flow data, predict the traffic flow characteristic $\boldsymbol{Y}$ over the next $T$ time intervals.*

### A. Graph Convolutional Neural Network

Let the values described traffic flow are obtained with the frequency $q$ times per day. Denote the current timestamp as $t_0$.

To take into account the current, daily and weekly patterns of the traffic flow distribution for each road link, we consider three time series. This approach was proposed in [26]:

1. The recent time series.
   To take into account the recent traffic flow values for traffic flow prediction, we consider the following time series that combines the feature vectors over the recent $T_c$ time intervals. :
   $$\boldsymbol{\chi}_c = \left(\mathbf{X}_{t_0 - T_h + 1}, \mathbf{X}_{t_0 - T_h + 2}, \cdots, \mathbf{X}_{t_0}\right). \tag{3}$$
2. The daily-periodic time series.
   Traffic flows have a daily-period component in traffic data (for example, morning and evening rush hours). To take into account the daily pattern, we consider the time series of the following form:
   $$\boldsymbol{\chi}_d = \left(\mathbf{X}_{t_0 - T_d * q}, \mathbf{X}_{t_0 - (T_d - 1) * q}, \cdots, \mathbf{X}_{t_0 - q}\right), \tag{4}$$
   where $T_d$ is the number of days to consider.
3. The weekly-periodic time series.
   The traffic flows distribution often varies depending on the day of the week (especially on weekdays/weekends). To take into account the weekly pattern, we consider the time series of the following form:
   $$\boldsymbol{\chi}_w = \left(\mathbf{X}_{t_0 - 7 * T_w * q}, \mathbf{X}_{t_0 - 7 * (T_w - 1) * q}, \cdots, \mathbf{X}_{t_0 - 7 * q}\right), \tag{5}$$
   where $T_w$ is the number of weeks to consider.

### B. Data Normalization

To increase the training speed,

The input feature vectors and the predicted values should be normalized to speed up the training of the neural network, improve convergence and reduce the probability of finding a local optimum. Data is normalized as follows:

$$\hat{x} = \frac{(x - x_{min})}{(x_{max} - x_{min})}, \tag{6}$$

where $x$ is the input data, $\hat{x}$ is the normalized data in the range $[0, 1]$.

### C. Graph Convolution in Spatial Dimension

The spectral graph theory generalizes the convolution operation from grid-based data to graph data. The graph convolution is designed to capture the topological properties of the graph-structured data to solve classification and regression tasks. In the considered problem, the road network is obviously a graph-structured data.

In the spectral graph theory, a graph is represented by its Laplacian matrix:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}, \tag{7}$$

or its normalized form:

$$\boldsymbol{L} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{\frac{1}{2}}, \tag{8}$$

where $\boldsymbol{A}$ is the adjacency matrix of the graph, $\boldsymbol{I}$ is a unit matrix, $\boldsymbol{D}$ is the matrix with the node degree values on the main diagonal:

$$\boldsymbol{D}_{ii} = \sum_j \boldsymbol{A}_{ij}. \tag{9}$$

The eigenvalue decomposition of the Laplacian matrix has the following form:

$$\boldsymbol{L} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T, \tag{10}$$

where $\boldsymbol{\Lambda}$ is a diagonal matrix, $\boldsymbol{U}$ is the Fourier basis.

The convolution operation on a graph is defined as the result of multiplying the signal $x\ in\ mathbbR^N$ on a graph with the kernel $g\ _{theta}$

The convolution operation on a graph is defined as the result of multiplying the signal $x \in \mathbb{R}^N$ on a graph with the kernel $g_\theta$ [28]:

$$g_\theta * x = g_\theta(\boldsymbol{L})\,x = g_\theta\left(\boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T\right)x = \boldsymbol{U}g_\theta(\boldsymbol{\Lambda})\,\boldsymbol{U}^T x. \tag{11}$$

As the input signal $x \in \mathbb{R}^N$ the $i$-th value of the feature vector $\mathbf{x}_t$ for all nodes of the graph at a specified time moment $t$ can be used.

However, the eigenvalue decomposition of the Laplacian matrix is a computationally difficult task. To approximately solve this problem, in [28] it was proposed to use Chebyshev polynomials :

$$g_\theta * x = g_\theta(\boldsymbol{L})\,x = \sum_{k=0}^{K-1} \theta_k T_k(\tilde{\boldsymbol{L}})x, \tag{12}$$

where the $\theta$ parameter is a vector of polynomial coefficients, $\tilde{\boldsymbol{L}} = \frac{2}{\lambda_{max}}\boldsymbol{L} - I$, $\lambda_{max}$ is the maximum eigenvalue of the Laplacian matrix. The Chebyshev polynomials are recursively defined as follows:

$$\begin{aligned} &T_k(x) = 2xT_{k-1}(x) - T_{k-2}(x), \\ &T_0(x) = 1, T_1(x) = x. \end{aligned} \tag{13}$$

As the activation function in the convolutional layer of graph neural network, the Rectified Linear Unit (ReLU) function is used.

### D. Convolution in Temporal Dimension

After performing the graph convolutional operations, the standard convolutional layer is embedded to perform a convolutional operation in the temporal dimension. Calculations on the $r$-th layer of the neural network can be written as:

$$\boldsymbol{\chi}^r = ReLU\left(\Phi * \left(ReLU\left(g_\theta * \boldsymbol{\chi}^{r-1}\right)\right)\right), \tag{14}$$

where $\Phi$ is the parameters of the convolutional kernel in the temporal dimension, $\boldsymbol{\chi}$ is the original input time series or the result of calculations on the previous layer of the neural network.

### E. Neural Network Architecture

The architecture of the used neural network is shown in Fig. 1. The network consists of three independent components with the same structure. Each component considers, respectively, the recent time series, the daily-periodic time series, and the weekly-periodic time series with different patterns in traffic data.

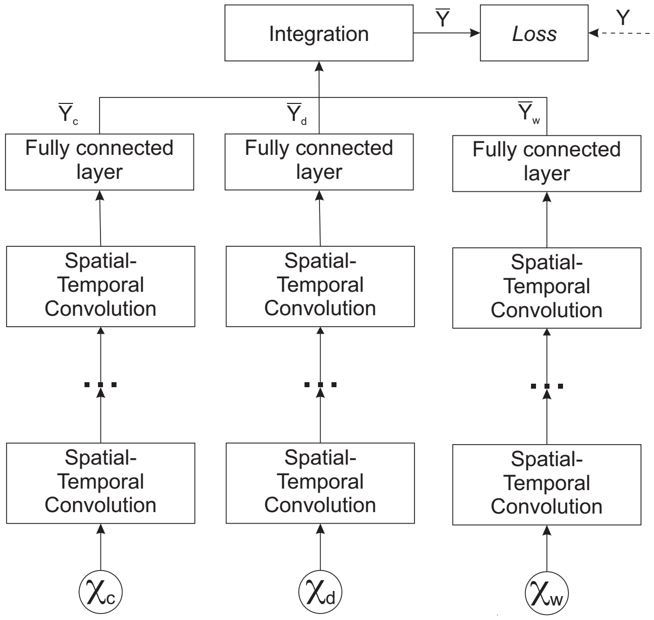


Fig. 1. Neural network architecture.

In summary, the neural network consists of several modules that perform spatio-temporal convolution on graph data that take into account spatio-temporal correlations in the traffic data. Several layers are performed sequentially to extract a larger range of spatio-temporal correlations. The last fully connected layer is necessary to obtain the required dimension of the output data of the neural network.

To combine the prediction results obtained from each of the three components (that uses recent, daily and weekly time series), the following operation is used:

$$\hat{\boldsymbol{Y}} = W_c \odot \hat{Y}_c + W_d \odot \hat{Y}_d + W_w \odot \hat{Y}_w, \tag{15}$$

where $\odot$ is the Hadamard product, $W_c, W_d, W_w$ are learning parameters.

## III. Experimental Analysis

### A. Dataset

Experimental studies of the model were carried out for the transportation network of the Samara city, Russia. To evaluate the accuracy and the computation time of the graph neural network model, an urban area including 1760 road segments was selected. As the traffic data for the experimental studies, the average traffic speed (in km/h) was used. The data set contains records for 60 days, from November 1, 2019. The average traffic speed was aggregating over a 10-minute time interval. The missing data values (for some road links and some time intervals) were filled using linear interpolation. The used dataset was divided into three parts: training set (60 % of the sample), control set (20 % of the sample) and validation set (20 % of the sample).

### B. Model Parameters

The developed model was implemented using the MXNet framework. Each convolutional layer contains 64 neurons. As a prediction horizon, we choose $T = 6$ time intervals, i.e. we predict traffic flow for the next hour.

The model parameters were learned using the training set, the best model was selected by the lowest error on the validation set. A comparison of models was carried out on the control set.

### C. Baseline Methods

Comparison of the graph convolutional neural networks (GCNN) was performed with the following baseline methods:

- Linear regression;
- Fully connected neural network with two hidden layers, 24 neurons on each layer (MLP);
- Recurrent neural network with long short-term memory architecture (LSTM).

Moreover, we test GCNN with a different number of Chebyshev polynomials $K \in \{1, 2, 3\}$.

To evaluate the accuracy of the models, three standard criteria were used: mean absolute error (MAE), root mean square error (RMSE), and mean absolute percentage error (MAPE).

### D. Results

Table I presents a comparison of the experimental results of the selected models for a prediction horizon of one hour.

TABLE I
MODELS ACCURACY COMPARISON

| | MAE, km/h | RMSE | MAPE, % |
|---|---|---|---|
| Linear regression | 1.96 | 2.98 | 8.53 |
| MLP | 1.69 | 2.73 | 7.12 |
| LSTM | 1.71 | 2.82 | 7.4 |
| GCNN, K=1 | 1.6538 | 2.72 | 7.02 |
| GCNN, K=2 | 1.6542 | **2.698** | 6.993 |
| GCNN, K=3 | **2.6515** | **2.698** | **6.987** |

Experimental analysis shows that the graph convolutional neural network investigated in this paper provides better prediction results than the baseline models. The best results according to the aggregate criterion were shown by the model with the third-order Chebyshev polynomials ($K = 3$).

At the next step, we compare the models for different prediction horizon values. The comparison results by the mean absolute error and the mean absolute percentage error are shown in Fig. 2 and Fig. 3 respectively.

Analyzing the obtained results, we can conclude that the presented model shows the best results for the entire prediction horizon, with an increase in the prediction horizon, the accuracy of all models decreases.

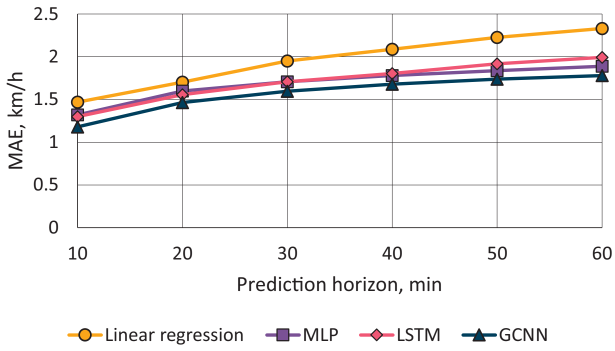


Fig. 2. Mean absolute error.

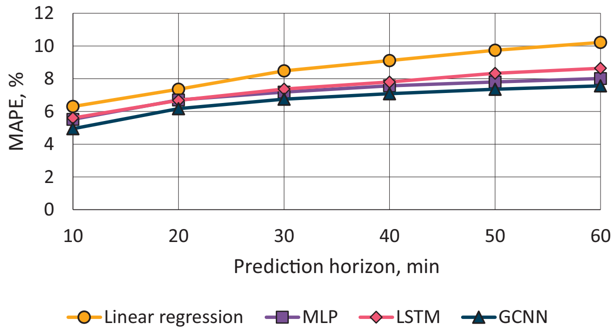


Fig. 3. Mean absolute percentage error.

At the final step, we evaluate the computation time of the graph neural network in comparison with the baseline models. We estimated the time for traffic flow prediction for all road network links for the next hour. Computations were performed using GPU nVidia GeForce GTX2080 Ti. The results are presented in Table II.

TABLE II
COMPUTATION TIME COMPARISON

| | MLP | GCNN, K=1 | GCNN, K=2 | GCNN, K=3 |
|---|---|---|---|---|
| Computation time, ms | 6.1 | 12.6 | 16.7 | 20.2 |

All considered models have a very low computation time and can be used for short-term traffic flow prediction in a large-scale transportation network.

## IV. CONCLUSIONS

In this paper, we investigate the graph convolutional neural network to solve the short-term traffic flow prediction problem. The considered model takes into account the spatial-temporal correlations in the traffic data and the topology of the road network.

Experimental studies conducted on a real dataset with the average traffic speed in the Samara city, Russia, have shown that the considered model has high prediction accuracy (average error about 7 %) and very low computation time (about 16-20 milliseconds) sufficient for traffic flow prediction in the real-time regime.

Future studies may be aimed at the development a more complicated model architecture and taking into account additional factors affecting the traffic distribution, for example, weather conditions or public events.

## REFERENCES


[1] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, pp. 1624–1639, 2011.

[2] A. A. Agafonov and V. V. Myasnikov, "Numerical route reservation method in the geoinformatic task of autonomous vehicle routing," Computer Optics, vol. 42, no. 5, pp. 912–920, 2018.

[3] E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," Transportation Research Part C: Emerging Technologies, vol. 43, pp. 3–19, 2014.

[4] I. Lana, J. Del Ser, M. Velez, and E. I. Vlahogianni, "Road Traffic Forecasting: Recent Advances and New Challenges," IEEE Intelligent Transportation Systems Magazine, vol. 10, no. 2, pp. 93–109, 2018.

[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] 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.

[7] S. Shekhar and B. M. Williams, "Adaptive seasonal time series models for forecasting short-term traffic flow," Transportation Research Record, no. 2024, pp. 116–125, 2007.

[8] S. R. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," Journal of Intelligent Transportation Systems: Technology, Planning, and Operations, vol. 13, no. 2, pp. 53–72, 2009.

[9] H. Su, L. Zhang, and S. Yu, "Short-term traffic flow prediction based on incremental support vector regression," presented at the Proceedings - Third International Conference on Natural Computation, ICNC 2007, 2007, vol. 1, pp. 640–645.

[10] Y.-S. Jeong, Y.-J. Byon, M. M. Castro-Neto, and S. M. Easa, "Supervised weighting-online learning algorithm for short-term traffic flow prediction," IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 4, pp. 1700–1707, 2013.

[11] D. Xia, B. Wang, H. Li, Y. Li, and Z. Zhang, "A distributed spatial-temporal weighted model on MapReduce for short-term traffic flow forecasting," Neurocomputing, vol. 179, pp. 246–263, 2016.

[12] A. A. Agafonov, A. S. Yumaganov, and V. V. Myasnikov, "Big data analysis in a geoinformatic problem of short-term traffic flow forecasting based on a K nearest neighbors method," Computer Optics, vol. 42, no. 6, pp. 1101–1111, 2018.

[13] A. Agafonov and A. Yumaganov, "Short-Term Traffic Flow Forecasting Using a Distributed Spatial-Temporal k Nearest Neighbors Model," in 2018 IEEE International Conference on Computational Science and Engineering (CSE), Bucharest, Romania, 2018, pp. 91–98.

[14] F. Moretti, S. Pizzuti, S. Panzieri, and M. Annunziato, "Urban traffic flow forecasting through statistical and neural network bagging ensemble hybrid modeling," Neurocomputing, vol. 167, pp. 3–7, 2015.

[15] W. Zheng, D.-H. Lee, and Q. Shi, "Short-term freeway traffic flow prediction: Bayesian combined neural network approach," Journal of Transportation Engineering, vol. 132, no. 2, pp. 114–121, 2006.

[16] R. Yu, Y. Li, C. Shahabi, U. Demiryurek, and Y. Liu, "Deep learning: A generic approach for extreme condition traffic forecasting," presented at the Proceedings of the 17th SIAM International Conference on Data Mining, SDM 2017, 2017, pp. 777–785.

[17] A. Agafonov and A. Yumaganov, "Bus Arrival Time Prediction with LSTM Neural Network," Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11554 LNCS, pp. 11–18, 2019.

[18] 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 (Switzerland), vol. 17, no. 4, 2017.

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

[20] [1]H. Yao et al., "Deep multi-view spatial-temporal network for taxi demand prediction," Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, 2018, pp. 2588–2595.

[21] S. Zhang, H. Tong, J. Xu, and R. Maciejewski, "Graph convolutional networks: a comprehensive review," Computational Social Networks, vol. 6, no. 1, 2019.

[22] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral Networks and Locally Connected Networks on Graphs," arXiv:1312.6203 [cs], May 2014, Accessed: May 11, 2020. [Online]. Available: http://arxiv.org/abs/1312.6203.

[23] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering," arXiv:1606.09375 [cs, stat], Feb. 2017, Accessed: May 11, 2020. [Online]. Available: http://arxiv.org/abs/1606.09375.

[24] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," arXiv:1609.02907 [cs, stat], Feb. 2017, Accessed: May 11, 2020. [Online]. Available: http://arxiv.org/abs/1609.02907.

[25] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, pp. 3634–3640, Jul. 2018.

[26] S. Guo, Y. Lin, N. Feng, C. Song, and H. Wan, "Attention Based Spatial-Temporal Graph Convolutional Networks for Traffic Flow Forecasting," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 922–929, Jul. 2019.

[27] Z. Cui, K. Henrickson, R. Ke, and Y. Wang, "Traffic Graph Convolutional Recurrent Neural Network: A Deep Learning Framework for Network-Scale Traffic Learning and Forecasting," IEEE Transactions on Intelligent Transportation Systems, pp. 1–12, 2019.

[28] D. K. Hammond, P. Vandergheynst, and R. Gribonval, "Wavelets on graphs via spectral graph theory," Applied and Computational Harmonic Analysis, vol. 30, no. 2, pp. 129–150, 2011.