Traffic Flow Forecasting Based on Adaptive Gated Graph Neural Network: Postprint

Authors: Wang Yang, Zheng Jin, Ying Liu, Li Ping

Date: 2022-04-07T15:01:57Z

Abstract

Traffic flow prediction constitutes a crucial component of intelligent transportation systems. Due to the complexity of traffic data, long-term and accurate traffic flow prediction has consistently remained one of the most challenging tasks in time series forecasting. In recent years, researchers have applied spatio-temporal graph modeling methods based on graph neural networks to traffic flow prediction tasks, achieving favorable predictive performance. However, existing graph modeling methods only reflect spatial dependencies in road networks through predefined adjacency structures, neglecting the importance of temporal correlation relationships between nodes for prediction. To address this limitation, we propose an Adaptive Gated Graph Neural Network (Ada-GGNN), whose core lies in simultaneously capturing the spatial structure of road networks and adaptive temporal correlations through a spatial transmission module, and learning time series features on nodes via a gating mechanism. Experimental results on two real-world traffic network datasets, PeMSD7 and Los-loop, demonstrate the superior performance of the proposed model.

Full Text

Preamble

Vol. 39 No. 8
Application Research of Computers
ChinaXiv Partner Journal

Traffic Flow Prediction Based on Adaptive Gated Graph Neural Network

Wang Yang, Zheng Jin†, Liu Ying, Li Ping
(College of Computer Science, Southwest Petroleum University, Chengdu 610500, China)

Abstract: Traffic flow prediction is a crucial component of intelligent transportation systems. Due to the complexity of traffic data, long-term and accurate traffic flow prediction has consistently been one of the most challenging tasks in time series forecasting. In recent years, researchers have applied spatio-temporal graph modeling methods based on graph neural networks to traffic flow prediction tasks, achieving promising performance. However, existing graph modeling methods only capture spatial dependencies in road networks through predefined adjacency structures, neglecting the importance of temporal correlations between nodes for prediction accuracy. To address this limitation, this paper proposes an Adaptive Gated Graph Neural Network (Ada-GGNN), whose core innovation lies in simultaneously capturing the spatial structure of road networks and adaptive temporal correlations through a spatial passing module, while learning temporal features on nodes via a gating mechanism. Experimental results on two real-world traffic network datasets, PeMSD7 and Los-loop, demonstrate the model' s superior performance.

Keywords: traffic flow prediction; spatio-temporal graph; adaptive gated graph neural networks; temporal correlation

0 Introduction

Intelligent transportation systems constitute an indispensable component of smart city development. As a foundational and critical research direction within these systems, accurate and real-time traffic flow prediction offers an effective solution for reducing traffic accidents, alleviating congestion, and improving transportation efficiency. In practice, traffic flow is typically recorded by sensors deployed on roads, capturing metrics such as volume, speed, and occupancy rates. The objective of traffic flow prediction is to forecast future traffic conditions based on historical data from these sensors across the road network. Traditional traffic flow prediction models primarily rely on statistical or machine learning approaches, including Historical Average (HA) [1], Autoregressive Integrated Moving Average (ARIMA) [2], Vector Autoregression (VAR) [3], and k-Nearest Neighbors (KNN) [4]. These methods often yield suboptimal predictions in real-world applications due to their reliance on stationarity assumptions. However, traffic conditions are inherently complex, exhibiting characteristics of randomness, periodicity, trend, and spatio-temporal dependence due to various environmental factors. Furthermore, these approaches require complex feature engineering and consider only temporal information while ignoring the critical importance of spatial information for traffic flow prediction.

In recent years, deep learning has been widely applied to traffic flow prediction tasks due to its powerful feature representation and nonlinear fitting capabilities. Yu et al. [5] utilized deep Long Short-Term Memory (LSTM) networks to predict traffic sequences, demonstrating superior results compared to traditional methods. However, this approach still fails to consider the spatial structure of traffic road networks. Shi et al. [6] designed the ConvLSTM model for precipitation prediction, which combines Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) to model spatial and temporal correlations respectively. Subsequently, Zhang et al. [7] converted traffic road networks into regular grids and employed CNNs to extract spatial features. While these models can capture spatio-temporal characteristics, traffic roads are essentially irregular structures, and treating traffic data as regular 2D or 3D grid data leads to loss of spatial topological information.

Researchers subsequently introduced Graph Neural Networks (GNN) to model spatio-temporal network data, achieving significant progress in traffic flow prediction. Li et al. [8] proposed the DCRNN model, which models traffic networks as directed graphs, employs bidirectional random walks to capture spatial dependencies between roads, and uses RNNs to capture temporal correlations. This approach simultaneously processes temporal and spatial information, yielding more accurate predictions. Following this paradigm, researchers have extensively explored combinations of GNN with CNN or RNN. For instance, references [9–11] all utilize GNN to capture spatial dependencies in topological structures while employing 1D convolutions or RNNs to capture dynamic temporal dependencies.

Despite the relative success of these graph neural network-based methods for traffic flow prediction, several limitations persist. First, these approaches only capture spatial dependencies between nodes through predefined graph adjacency structures, overlooking the temporal correlations between nodes. Predefined adjacency matrices are typically constructed based on distances between sensor locations or upstream-downstream relationships, which can result in situations where two nodes exhibit correlated time series but lack direct connections. As illustrated in Figure 1, the left diagram shows node distribution in a traffic road network, while the right diagram shows solid lines representing predefined adjacency structures and dashed lines indicating node pairs with temporal correlations. The figure demonstrates that relying solely on predefined adjacency relationships cannot comprehensively reflect mutual influences between nodes. Second, another limitation of existing graph neural network methods for traffic flow prediction is that they often explore temporal and spatial features separately before fusing them, a delayed fusion strategy that prevents the model from accessing global information in real-time, thereby affecting prediction performance.

Figure 1. Problem description

To address these issues, this paper proposes a novel end-to-end spatio-temporal data prediction model called Ada-GGNN. The model first captures spatial dependencies in traffic road networks through a spatial passing module while simultaneously mining temporal correlations between nodes. It then employs GRU [12] to learn dynamic temporal dependencies in traffic data and performs multiple spatio-temporal fusions at each time step to explore high-order interactions between time and space, obtaining spatio-temporal embedding representations.

Finally, these representations are passed through fully connected layers to produce predictions. The main contributions of this work are as follows:

1. An adaptive correlation matrix between time series is learned in a data-driven manner to capture associations between nodes that are lost due to predefined adjacency structures.
2. A cyclic aggregation method is adopted to fuse spatio-temporal embedding representations in real-time, enriching the model' s spatio-temporal embeddings.
3. Experimental results on two real-world traffic datasets demonstrate the effectiveness of Ada-GGNN. The proposed method also exhibits strong generalization and can be easily extended to other spatio-temporal data mining tasks.

1.1 Graph Neural Networks

Gori et al. [13] first proposed Graph Neural Networks (GNN) for graph-structured data. Subsequent improvements and variants of GNN have achieved remarkable progress across various tasks. Defferrard et al. [14] improved GCN using fast localized convolutional filters, proposing the ChebNet model. Kipf et al. [15] simplified ChebNet using a first-order approximation of spectral convolution, achieving excellent classification performance in semi-supervised tasks. Reference [16] improved scalability for large graphs by sampling a fixed number of neighbors for each node and aggregating their features. Reference [17] proposed the GAT model, a powerful GCN variant that introduces an attention mechanism to dynamically determine the importance of each neighbor node to the central node. However, these GNN models typically update node hidden states using multi-layer perceptrons, which limits long-range information propagation in graphs.

To address this limitation, Li et al. [18] proposed Gated Graph Neural Networks (GGNN), a classic spatial domain message-passing model based on GRU. Due to its effectiveness on spatio-temporal data, GGNN has been applied to various tasks. Zhang et al. [19] proposed the TextING model, which constructs individual graphs for each document and uses GGNN to learn word node embeddings, demonstrating superiority over state-of-the-art text classification methods through extensive experiments. Reference [20] focused on a novel financial event prediction task, using GGNN to update event representations based on event graphs. Reference [21] improved GGNN to better model node interactions and infer compatibility from graphs. Additionally, GGNN has been applied to other tasks such as recommendation [22], image classification [23], and situation recognition [24]. For spatio-temporal data prediction problems, GGNN has demonstrated excellent performance. Inspired by this, we adopt the GGNN framework with a learnable adaptive node correlation matrix to address both short-term and long-term traffic prediction challenges.

2.1 Problem Definition

In traffic prediction tasks, each sensor is typically treated as a node, with distances between sensors or upstream-downstream relationships forming edges between nodes. The traffic network can thus be defined as an undirected graph $G = (V, E, A)$, where V is the set of nodes, E is the set of edges, and $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of graph G. At time step t, the traffic conditions of N nodes are represented as $X_t \in \mathbb{R}^{N\times D}$, where D denotes the feature dimension.

Therefore, the traffic prediction problem can be formulated as learning a mapping function f that, given a road network topology and historical traffic information over T time steps, can predict traffic information for the next τ moments, as defined in Equation (1):

$$\hat{X}_{t+1}, \dots, \hat{X}_{t+\tau} = f(G; X_{t-T+1}, \dots, X_t)$$

2.2 Model Architecture

The Ada-GGNN model primarily consists of two-stage spatio-temporal fusion modules at each time step. Each spatio-temporal fusion module comprises two components: a Spatial Passing (SP) module and a Gated Recurrent Unit (GRU). As shown in Figure 2, the model first uses historical data from T time steps as input. At each time step, the spatial passing module captures both the fixed topology of the traffic road network and learns adaptive correlations between sequences. The time series with spatial features are then fed into the GRU, which captures dynamic changes in sequential data, extracts temporal features, and performs spatio-temporal data fusion. Finally, a fully connected layer produces predictions for τ time steps.

Figure 2. Model architecture diagram

2.3 Spatial Passing Module

Considering the topology of traffic road networks, the spatial passing module primarily captures spatial dependency relationships between roads. As shown in Figure 3, the module consists of two parts: one obtains fixed spatial relationships through a predefined adjacency matrix, while the other captures correlations between nodes through a learnable correlation matrix. Graph convolution operations are employed to extract information transmitted between nodes. For explicit spatial relationships in traffic networks, the adjacency matrix is typically constructed based on positional distances between sensors, denoted as $A_{org} \in \mathbb{R}^{N\times N}$. Let X_t represent the input traffic data, $W_{org} \in \mathbb{R}^{D\times F}$ represent the model parameter matrix, and ReLU activation function is adopted. This graph convolutional layer is defined as in Equation (2):

$$Z_{t,k}^{org} = \mathrm{ReLU}(A_{org} X_t W_{org}), \quad k \in [1,2]$$

To mine temporal correlations between node sequences in traffic networks, let $A_{ada} \in \mathbb{R}^{N \times N}$ denote the learnable adaptive correlation matrix between nodes, randomly initialized and automatically capturing adjusting correlation strengths during training. This graph convolutional layer is defined as in Equation (3):

$$Z_{t,k}^{ada} = \mathrm{ReLU}(A_{ada} X_t W_{ada}), \quad k \in [1,2]$$

Notably, this module performs second-order interactions on the topology of traffic networks ($k \in [1,2]$). When $k = 1$, $I = X_t$; when $k = 2$, $I = Z_{t,k=1}^{ada}$ (as shown in Figure 1). Finally, information extracted from both components is fused via concatenation, as in Equation (4):

$$Z_{t,k} = \mathrm{Concat}(Z_{t,k}^{org}, Z_{t,k}^{ada})$$

Figure 3. The architecture of the SP module

2.4 Spatio-Temporal Fusion Module

The spatial passing module captures spatial features and sequence correlations between nodes at each time step and extracts their fused feature representations. However, traffic prediction is a spatio-temporal task that requires capturing not only spatial information but also temporal dependencies in sequences. Therefore, in the spatio-temporal fusion module, GRU captures temporal dependencies along the sequence dimension, while cyclic aggregation enables real-time spatio-temporal fusion at each time step. As shown in Equation (5), the GRU receives two inputs: the current time step's data embedding with spatial information $Z_{t,k}$ and the previous state representation $H_{t,k}^{pre}$. Since Ada-GGNN adopts second-order spatio-temporal interactions at each time step (as shown in Figure 1), the previous state expression differs at order k, as specified in Equation (6):

$$H_{t,k} = \mathrm{GRU}(Z_{t,k}, H_{t,k}^{pre}), \quad k \in [1,2]$$

$$H_{t,k}^{pre} = \begin{cases} H_{t-1,k+1} & k = 1 \\ H_{t,k-1} & k = 2 \end{cases}$$

GRU controls the degree to which current input and previous state are written through gating mechanisms. Larger update gate values and smaller reset gate values indicate less information from the previous state being written. In this manner, Ada-GGNN can capture long-term temporal correlations in traffic data. The detailed operations are shown in Equations (7)-(11):

$$u_{t,k} = \sigma(W_u Z_{t,k} + b_u + W^u_{pre} H^{pre}_{t,k} + b^u_{pre})$$

$$r_{t,k} = \sigma(W_r Z_{t,k} + b_r + W^r_{pre} H^{pre}_{t,k} + b^r_{pre})$$

$$\tilde{h}_{t,k} = \tanh(W_h Z_{t,k} + b_h + r_{t,k} \odot (W^h_{pre} H^{pre}_{t,k} + b^h_{pre}))$$

$$H_{t,k} = (1 - u_{t,k}) \odot H^{pre}_{t,k} + u_{t,k} \odot \tilde{h}_{t,k}$$

where $u_{t,k}$ and $r_{t,k}$ represent the update gate and reset gate sizes respectively, σ denotes the Sigmoid activation function, and W and b are learnable parameters.

After T time steps of spatio-temporal fusion, Ada-GGNN employs a fully connected layer to obtain final predictions $\hat{Y}_{out}$, as in Equation (12):

$$\hat{Y}_{out} = W_{out} H_{t=T,k=2} + b_{out}$$

Considering that traffic prediction is a regression task, Mean Absolute Error (MAE) is adopted as the loss function, defined in Equation (13):

$$\mathrm{MAE} = \frac{1}{M} \sum_{i=1}^{M} |y_i - \hat{y}_i|$$

2.5 Prediction

The prediction section details the final output mechanism of the model.

3.1 Datasets

To evaluate the proposed model' s effectiveness, experiments were conducted on two publicly available real-world traffic datasets: PeMSD7 and Los-loop. The PeMSD7 dataset contains traffic speed data collected by 228 sensors in California during May and June 2012 (weekdays only, 44 days total). The Los-loop dataset comprises traffic speed data collected by 207 detectors on Los Angeles highways from March 1-7, 2012. In all experiments, traffic speeds aggregated at 5-minute intervals were used as features, with the first 80% of data selected as the training set and the remaining 20% as the test set. Detailed statistics of the datasets are presented in Table 1.

Table 1. Summary statistics of two traffic network datasets

Dataset	Nodes	Edges	Max Value	Mean Value
PeMSD7				
Los-loop				

3.2 Experimental Settings

The model was implemented using Python and the PyTorch deep learning framework. The input consisted of 60 minutes of historical traffic data to predict traffic speeds for the next 15, 30, and 60 minutes. During training, the learning rate was set to 0.001, epochs to 1000, and the Adam optimizer [25] was employed. Early stopping was used to prevent overfitting. All experiments were conducted on a single NVIDIA Tesla T4 GPU with 16GB RAM.

The proposed model was compared against seven baseline methods:

a) **ARIMA**: Autoregressive Integrated Moving Average, a classical autoregressive model combining autoregression, moving average, and differencing for time series prediction.

b) **SVR** [26]: Support Vector Regression, a key application branch of SVM. Linear kernel functions were adopted for traffic prediction tasks.

c) **ASTGCN** [10]: This model consists of three independent components modeling hourly, daily, and weekly temporal attributes, each containing spatio-temporal attention mechanisms and spatio-temporal convolution operations, with final predictions generated through weighted fusion.

d) **T-GCN** [11]: This model uses GCN to learn traffic network topology and GRU to learn dynamic changes in traffic data, thereby capturing spatial and temporal dependencies.

e) **A3T-GCN** [27]: Built upon T-GCN, this model introduces attention mechanisms to adjust the importance of different time points and assemble global temporal information for traffic prediction.

f) **LSGCN** [28]: This model integrates GCN and graph attention networks into a spatial gating block, using spatial gating blocks and Gated Linear Units [29] (GLU) to capture spatio-temporal features for prediction.

g) **GWN** [30]: This model captures hidden spatial dependencies in data through a learnable approach and uses dilated 1D convolution components to expand the receptive field for processing long sequences.

To verify Ada-GGNN’ s effectiveness, three evaluation metrics were employed:

a) **Root Mean Square Error (RMSE)**: $\text{RMSE} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(y_i - \hat{y}_i)^2}$

b) **Mean Absolute Error (MAE)**: $\text{MAE} = \frac{1}{M}\sum_{i=1}^{M}|y_i - \hat{y}_i|$

c) **Mean Absolute Percentage Error (MAPE)**: $\text{MAPE} = \frac{1}{M}\sum_{i=1}^{M}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%$

where M represents the number of samples, y_i denotes the actual traffic speed of the i-th sample, and $\hat{y}_i$ represents the model' s predicted value.

3.3 Experimental Results Analysis

Table 2 presents performance comparisons between Ada-GGNN and other baseline models for 15, 30, and 60-minute traffic predictions on both datasets. The following observations can be drawn:

Neural network-based methods achieve lower prediction errors than traditional time series analysis and machine learning methods (e.g., ARIMA and SVR) across all evaluation metrics on both datasets. This is because ARIMA and SVR have limited nonlinear modeling capabilities. Particularly, ARIMA requires high data stability, but traffic data is complex with numerous influencing factors, resulting in the least favorable predictions. In contrast, neural network models not only excel at modeling nonlinear data but also incorporate topological structures to obtain spatial features of traffic networks, thereby demonstrating superior performance.

Ada-GGNN achieves state-of-the-art prediction performance compared to other baseline models for both long-term and short-term predictions on both datasets. For instance, for the 15-minute prediction task on the Los-loop dataset, Ada-GGNN improves upon GWN by 2.21% and LSGCN by 8.10% in terms of RMSE. Similar results are observed across other time horizons and evaluation metrics. These improvements are primarily attributed to Ada-GGNN' s architectural design. First, Ada-GGNN outputs predictions for multiple time steps simultaneously, whereas LSGCN must generate predictions sequentially based on previous outputs, leading to error accumulation and inferior long-term performance. Second, Ada-GGNN employs a learnable approach to mine correlations between node sequences in traffic networks, enriching node embeddings with new useful information that enhances prediction accuracy. Finally, the proposed model applies cyclic aggregation to learn spatio-temporal embeddings through real-time fusion rather than separate extraction and delayed fusion, facilitating the learning of fine-grained dynamic spatio-temporal relationships.

Table 2. Performance comparison of Ada-GGNN and other baselines on PeMSD7 and Los-loop datasets

Datasets	Models	15min	30min	60min
		RMSE	MAE	MAPE
PeMSD7	ARIMA	12.29	10.12	25.29%
	SVR	4.89	4.08	8.28%

Datasets	Models	15min	30min	60min
	ASTGCN	3.41	1.94	4.50%
	T-GCN	3.45	1.93	4.41%
	A3T-GCN	3.40	1.89	4.30%
	LSGCN	3.37	1.92	4.33%
	GWN	3.39	1.89	4.32%
	Ada-GGNN	**3.30**	**1.82**	**4.21%**
Los-loop	ARIMA	10.05	7.71	20.84%
	SVR	5.65	3.86	9.33%
	ASTGCN	5.25	3.27	8.05%
	T-GCN	5.21	3.27	8.34%
	A3T-GCN	5.12	3.17	8.08%
	LSGCN	5.31	3.35	8.33%
	GWN	4.99	2.86	7.32%
	Ada-GGNN	**4.88**	**2.79**	**7.16%**

3.3.1 Influence of Hidden Layer Units

The number of hidden layer units is often a direct cause of overfitting, making its configuration crucial during model tuning. This experiment selects hidden unit quantities from [16, 32, 64, 96, 128, 160] to analyze their impact on prediction accuracy. As shown in Figure 4, the x-axis represents the number of hidden units, while the y-axis shows different metrics. The results indicate that as the number of hidden units increases, RMSE and MAE first decrease and then increase, with the lowest prediction errors achieved when the hidden layer contains 96 units for both Los-loop and PeMSD7 datasets. This is primarily because excessive hidden units increase model complexity and risk of overfitting.

Figure 4. Influence of different hidden units on both datasets

3.3.2 Model Robustness Verification

During traffic data collection, sensors inevitably introduce noise into the data. Therefore, perturbation analysis experiments were conducted to test Ada-GGNN' s robustness. As shown in Figure 5, two types of random noise were added: Gaussian noise and Poisson noise. Gaussian noise follows a normal distribution $\mathcal{N}(\mu = 0, \sigma \in [0.2, 0.4, 0.6, 0.8, 1])$; Poisson noise follows a Poisson distribution $\mathcal{P}(\lambda \in [1, 2, 4, 8, 16])$. The results demonstrate that Ada-GGNN exhibits minimal changes in evaluation metrics under both noise distributions, indicating strong adaptability to noisy data and robust performance.

Figure 5. The influence of Gaussian perturbation and Poisson perturbation on the Ada-GGNN on the Los-loop dataset

3.3.3 Validation of Adaptive Correlation Matrix

To validate the effectiveness of adaptive correlation matrix learning in the spatial passing module, a comparative experiment was designed. The No_Ada model removes adaptive temporal correlation learning from Ada-GGNN, relying solely on the predefined adjacency matrix to capture spatial dependencies. Experimental results for 15-minute and 60-minute prediction tasks on both datasets are shown in Table 3.

Table 3. Influence of learning time-series correlation in the SP module

Datasets	Models	15min	60min
		RMSE	MAE
PeMSD7	No_Ada	3.59	1.99
	Ada-GGNN	**3.30**	**1.82**
Los-loop	No_Ada	5.05	2.86
	Ada-GGNN	**4.88**	**2.79**

The results clearly show that Ada-GGNN significantly outperforms No_Ada across different tasks and datasets. This is because Ada-GGNN captures not only spatial relationships in road networks but also learns temporal correlations between nodes, compensating for the limitation of previous graph neural network-based traffic flow prediction methods that contained only single spatial relationships.

To further interpret the role of the learned adaptive correlation matrix, additional experiments were conducted, visualizing geographic locations of partial nodes from the Los-loop dataset alongside heatmaps of the learned correlation matrix. Figure 6(a) shows the geographic locations of six nodes, while Figure 6(b) displays the normalized correlation matrix heatmap between these six node sequences. First, Figure 6(b) clearly shows that column 2 has more high-value points than other columns, indicating that node 2 has greater influence on other nodes during prediction and exhibits higher correlations with other node sequences. Second, in the predefined fixed adjacency matrix, node 6 has no connection with nodes 3 and 4 due to large distances. However, Figure 6(b) reveals that node 6 significantly influences and correlates with nodes 3 and 4. This demonstrates that Ada-GGNN can discover node connections lost in predefined adjacency matrices due to distance constraints, introducing new useful information that improves prediction accuracy.

Figure 6. The learned self-adaptive correlation matrix

3.3.4 Model Computational Cost

Table 4 presents computational time comparisons between Ada-GGNN and other neural network baselines on the Los-loop dataset. During training, Ada-GGNN is 1.07 seconds faster per epoch than ASTGCN and 8.65 seconds faster than GWN, but approximately 2-3 seconds slower than T-GCN, A3T-GCN, and LSGCN. This is primarily because T-GCN uses only one graph convolutional layer and one GRU for flow prediction, while GWN has a more complex model structure with residual connections in each spatio-temporal layer, resulting in greater computational overhead. On the test set, LSGCN requires more time than other models because it must generate predictions sequentially based on previous outputs, whereas other models produce multiple time step predictions simultaneously. Considering both prediction accuracy and computational cost, Ada-GGNN demonstrates strong overall performance.

Table 4. Computation time on the Los-loop dataset

Models	Training (s/epoch)	Testing (s)
ASTGCN		
T-GCN		
A3T-GCN		
LSGCN		
Ada-GGNN		

4 Conclusion

This paper proposes a novel end-to-end Adaptive Gated Graph Neural Network (Ada-GGNN) model for traffic flow prediction. The model utilizes a learnable adaptive matrix to capture sequence correlations between nodes from data, introducing new useful information, and employs a gated graph neural network framework to handle long-term prediction problems. Ada-GGNN achieves the best results on two public traffic datasets. Considering that real-world traffic prediction is influenced by many external factors such as weather, temperature, and social events, future work will incorporate more additional features as input and explore scalable methods for applying Ada-GGNN to large graphs.

References

[1] Liu Jing, Guan Wei. A summary of traffic flow forecasting methods [J]. Journal of Highway and Transportation Research and Development, 2004, 21 (3): 82-85.

[2] Williams B M, Hoel L A. Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: theoretical basis and empirical results [J]. Journal of Transportation Engineering, 2003, 129 (6): 664-672.

[3] Zivot E, Wang Jiahui. Vector autoregressive models for multivariate time series [M]// Modeling Financial Time Series with S-Plus®. New York, NY, USA: Springer, 2003: 369–413.

[4] Zhang Xiaoli, He Guoguang, Lu Huapu. Short-term traffic flow forecasting based on K-nearest neighbors non-parametric regression [J]. Journal of Systems Engineering, 2009, 24 (2): 178-183.

[5] Yu R, Li Yaguang, Shahabi C, et al. Deep learning: a generic approach for extreme condition traffic forecasting [C]// Proc of SIAM International Conference on Data Mining. 2017: 777-785.

[6] Shi Xingjian, Chen Zhourong, Wang Hao, et al. Convolutional LSTM network: a machine learning approach for precipitation nowcasting [C]// Proc of the 28th International Conference on Neural Information Processing Systems. Cambridge, MA: MIT Press, 2015: 802-810.

[7] Zhang Junbo, Zheng Yu, Qi Dekang. Deep spatio-temporal residual networks for citywide crowd flows prediction [C]// Proc of the 31st AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2017: 1655-1661.

[8] Li Yaguang, Yu R, Shahabi C, et al. Diffusion convolutional recurrent neural network: data-driven traffic forecasting [C/OL]/ Proc of the 6th International Conference on Learning Representations. [S. l.] : OpenReview. net, 2018. (2018) [2022-01-12]. https://openreview. net/forum?id=SJiHXGWAZ.

[9] Yu Bing, Yin Haoteng, Zhu Zhanxing. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting [C]// Proc of the 27th International Joint Conference on Artificial Intelligence. California: IJCAI, 2018: 3634-3640.

[10] Guo Shengnan, Lin Youfang, Feng Ning, et al. Attention based spatial-temporal graph convolutional networks for traffic flow forecasting [C]// Proc of the 33rd AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019: 922-929.

[11] Zhao Ling, Song Yujiao, Zhang Chao, et al. T-GCN: a temporal graph convolutional network for traffic prediction [J]. IEEE Trans on Intelligent Transportation Systems, 2020, 21 (9): 3848-3858.

[12] Cho K, Van Merrienboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation [C]// Proc of EMNLP. Stroudsburg, PA: ACL, 2014: 1724-1734.

[13] Gori M, Monfardini G, Scarselli F. A new model for learning in graph domains [C]// Proc of IEEE IJCNN. Piscataway, NJ: IEEE Press, 2005: 729-734.

[14] Defferrard M, Bresson X, Vandergheynst P. Convolutional neural networks on graphs with fast localized spectral filtering [C]// Proc of the 30th International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc, 2016: 3837-3845.

[15] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks [C/OL]/ Proc of the 5th International Conference on

Learning Representations. [S. l.] : OpenReview. net, 2017. [2022-01-12]. https://openreview. net/forum?id=SJU4ayYgl.

[16] Hamilton W L, Ying R, Leskovec J. Inductive representation learning on large graphs [C]// Proc of the 31st International Conference on Neural Information Processing Systems. Red Hook, NY: Curran Associates Inc, 2017: 1025-1035.

[17] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks [C/OL]/ Proc of the 6th International Conference on Learning Representations. [S. l.] : OpenReview. net, 2018. (2018) [2022-01-12]. https://openreview. net/forum?id=rJXMpikCZ.

[18] Li Yujia, Zemel R, Brockschmidt M, et al. Gated graph sequence neural networks [C/OL]/ Proc of the 4th International Conference on Learning Representations. 2016. (2016) [2022-01-12]. http://arxiv. org/abs/1511. 05493.

[19] Zhang Yufeng, Yu Xueli, Cui Zeyu, et al. Every document owns its structure: inductive text classification via graph neural networks [C]// Proc of the 58th Annual Meeting of the Association for Computational Linguistics. Stroudsburg, PA: ACL, 2020: 334-339.

[20] Yang Yiying, Wei Zhongyu, Chen Qin, et al. Using external knowledge for financial event prediction based on graph neural networks [C]// Proc of the 28th ACM International Conference on Information and Knowledge Management. New York, NY, USA: ACM Press, 2019: 765-774.

[21] Cui Zeyu, Li Zekun, Wu Shu, et al. Dressing as a whole: outfit compatibility learning based on node-wise graph neural networks [C]// Proc of WWW Conference. New York, NY, USA: ACM Press, 2019: 307-317.

[22] Wu Shu, Tang Yuyuan, Zhu Yanqiao, et al. Session-based recommendation with graph neural networks [C]// Proc of the 33rd AAAI Conference on Artificial Intelligence. Palo Alto, CA: AAAI Press, 2019: 346-353.

[23] Marino K, Salakhutdinov R, Gupta A. The more you know: using knowledge graphs for image classification [C]// Proc of IEEE CVPR. Los Alamitos, CA: IEEE Computer Society, 2017: 20-28.

[24] Li Ruiyu, Tapaswi M, Liao Renjie, et al. Situation recognition with graph neural networks [C]// Proc of IEEE ICCV. Los Alamitos, CA, USA: IEEE Computer Society, 2017: 4183-4192.

[25] Kingma D P, Ba J. Adam: a method for stochastic optimization [C/OL]/ Proc of the 3rd International Conference on Learning Representations. 2015. (2015) [2022-01-12]. http://arxiv. org/abs/1412. 6980.

[26] Smola A J, Schölkopf B. A tutorial on support vector regression [J]. Statistics and Computing, 2004, 14 (3): 199-222.

[27] Bai Jiandong, Zhu Jiawei, Song Yujiao, et al. A3T-GCN: attention temporal graph convolutional network for traffic forecasting [J]. ISPRS International

Journal of Geo-Information, 2021, 10 (7): 485.

[28] Huang Rongzhou, Huang Chuyin, Liu Yubao, et al. LSGCN: long short-term traffic prediction with graph convolutional networks [C]// Proc of the 29th International Joint Conference on Artificial Intelligence. California: IJCAI, 2020: 2355-2361.

[29] Dauphin Y N, Fan A, Auli M, et al. Language modeling with gated convolutional networks [C]// Proc of the 34th International Conference on Machine Learning. [S. l.] : PMLR, 2017: 933-941.

[30] Wu Zonghan, Pan Shirui, Long Guodong, et al. Graph WaveNet for deep spatial-temporal graph modeling [C]// Proc of the 28th International Joint Conference on Artificial Intelligence. California: IJCAI, 2019: 1907-1913.

Note: Figure translations are in progress. See original paper for figures.