TAGnn: Time Adjoint Graph Neural Network for Traffic Forecasting

Qi Zheng and Yaying Zhang(✉)

The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 200092, China
{zhengqi97,yaying.zhang}@tongji.edu.cn

Abstract. Spatial-temporal modeling considering the particularity of traffic data is a crucial part of traffic forecasting. Many methods take efforts into relatively independent time series modeling and spatial mining and then stack designed space-time blocks. However, the intricate deep structures of these models lead to an inevitable increment in the training cost and the interpretability difficulty. Moreover, most previous methods generally ignore the meaningful time prior and the spatial correlation across time. To address these, we propose a novel Time Adjoint Graph Neural Network (TAGnn) for traffic forecasting to model entangled spatial-temporal dependencies in a concise structure. Specifically, we inject time identification (i.e., the time slice of the day, the day of the week) which locates the evolution stage of traffic flow into node representation. Secondly, based on traffic propagation, we connect data across time slices to generate the time-adjoint hidden feature and spatial correlation matrix, allowing the spatial-temporal semantics to be captured by a simple graph convolution layer. And we introduce a time residual connection in generating predictions to capture the future traffic evolution. Experiments on four traffic flow datasets demonstrate that our method outperforms the state-of-the-art baselines efficiently.

Keywords: Spatial-temporal mining · Graph neural network · Traffic forecasting

1 Introduction

Real-time and dynamic traffic forecasting is vital in the growing demand for intelligent transportation services. Accurate traffic prediction greatly impacts on urban spatial-temporal situation awareness for city management and travel planning. As a widely concerned spatial-temporal data forecasting problem in both academia and industry, traffic prediction has its uniqueness due to the following temporal and spatial observations:

(1) **The time-prior information helps to locate the evolution of traffic dynamics in periodic changes.** Traffic flow naturally has its unique temporal characteristics. Flow series recorded in road sensors always show similar periodic changes on daily and weekly scales. Intuitively, given a road section

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13943, pp. 369–379, 2023.
https://doi.org/10.1007/978-3-031-30637-2_24

with its historical traffic values and the time to predict (e.g. the peak hour on a weekday), a basic inference can be carried out easily. The time-prior information implies the innate traffic representation based on long-term observations. It could be beneficial in the traffic prediction tasks to directly use this intact and readily available time information (e.g. the time slice of the day, the day of the week) without much additional acquisition cost. Besides, based on the general temporal cycle pattern, modeling the change amount of traffic flow instead of the original value in a short future evolution may be easier for forecasting.

(2) **The influence range of spatial dependence (i.e. local to global) between road sensors varies according to the time span of traffic flow propagation.** As traffic state is transmitted along the road structure with time, traffic information of one location directly affects its spatially nearby neighbors in a short-term period (e.g. 5 min). However, when a traffic state (e.g. congestion) lasts a relatively long time, the scope of affected spatial locations will become wider. This means that the traffic status of one location can potentially influence that of distant locations over time as well as reflect the influence of historical states of locations far away from it. The closer two traffic locations are, the more often the spatial influence between them occurs within a short period. Conversely, spatial influence at more distant locations tends to persist over time due to traffic propagation [14]. How to extract global spatial connections related to time spans is nontrivial.

In recent years, deep learning-based models have been continuously proposed for capturing the temporal and spatial correlations in spatial-temporal series data. Many researchers utilize widely-used time-related models (e.g. Recurrent Neural Networks, Temporal Convolution Networks, and Transformer-liked models) to capture time dependencies. However, the physical time label describing the evolution stage of traffic flow series is usually neglected in these methods. Studies based on spatial mining made progress by considering varied adjacency relationships in short-term time slices but they generally ignore direct spatial influences that exist on different time spans. It is still a challenge to appropriately capture the entangled spatial and temporal dependencies in different space-time scales. Besides, the separate capture of temporal and spatial features hinders the exploration the space-time interaction in traffic propagation.

To address these issues, a Time Adjoint Graph neural network (TAGnn) for traffic forecasting is proposed in this work. The proposed model TAGnn can explicitly use the time-prior to increase the accuracy and reliability of prediction and dynamically mine the spatial-temporal dependencies from different space-time scales. The main contributions of this work are as follows:

(1) A new time encoding method is designed to make explicit use of time-prior information in the node representation, which provides considerable gains in accuracy. Besides, the time-residual connection is introduced in the generation of predicted values to capture the evolution of traffic flow.

(2) The spatial-temporal characteristics in each time slice are captured through novel yet simple spatial-temporal mining modules, which connect elements of input series from different time spans to extract spatial dependencies across time. It is conducive to exploring spatial-temporal interaction on appropriate space-time scales.

(3) Extensive experiments on four public real-world traffic datasets show that the proposed method outperforms state-of-the-art baselines with less training time cost.

2 Related Work

Time analysis and spatial mining are two key parts of the traffic forecasting problem. Early methods [8,15] are computationally efficient but perform poorly in complex scenarios. RNN-based, CNN-based and Transformer-based [10] models [2,5,6,11,12] can extract short-term and long-term temporal correlations in time series. Some other methods [4,9,14] capture cross-time dependencies. However, these models devoted to time dependence extraction usually rely only on the implied time characteristics or location coding of input data itself, and rarely take into account the auxiliary gain of time prior information. As advanced research on spatial mining, an increasing number of methods are focusing on the non-local and dynamic spatial influence between road nodes [2,11,13]. However, it is still challenging to explore the global spatial-temporal interaction in an appropriate space and time scale. Two recent studies [7,14] have turned the spotlight on mining data connections across different time horizons. They put forward some new ideas about modeling spatial-temporal interactions with different mechanisms, but there may be redundancy in the construction of adjacency matrices and input representations.

3 Preliminary

A group of road sensors distributed in a road network can be formulated as a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is a set of nodes meaning the road sensors, $|\mathcal{V}| = N$. $\mathcal{E}$ is a set of edges. In the urban road network, the 24-h traffic data collected by sensors are generally aggregated at a certain frequency (e.g., once every 5 min), so that a day can be divided into multiple time slices (e.g., 288). The traffic data collected by one sensor $v_i \in \mathcal{V}$ is a time sequence composed of multiple discrete values $\{x_{v_i;t_1}, x_{v_i;t_2}, ...||x_{v_i;t_j} \in \mathbb{R}^C\}$, and C is the number of features. At a certain time slice t_i, traffic features of all N graph nodes form the feature matrix $X_{t_i} \in \mathbb{R}^{N\times C}$. The traffic forecasting problem on a road network $\mathcal{G}$ can be formulated as: Given historical observation $\mathbf{X} = [X_{t-p}, ..., X_{t-1}, X_t]$ in the past P time slices (here $p = P-1$ for brevity), we aim to predict the traffic data $\mathbf{Y} = [X_{t+1}, ..., X_{t+2}, X_{t+Q}]$ in the next consecutive Q time slices. Here, time slice t denotes the most recent slice in the input series.

4 Methodology

The core idea of the proposed model TAGnn is to inject the time-prior information into node representation and connect data in different time spans to produce the time-adjoint hidden feature and the across-time global spatial correlation. Then the spatial-temporal feature can be captured by a simple graph convolution layer in each time slice. And a time-residual connection is introduced to generate predicted values for capturing the future evolution of traffic flow.

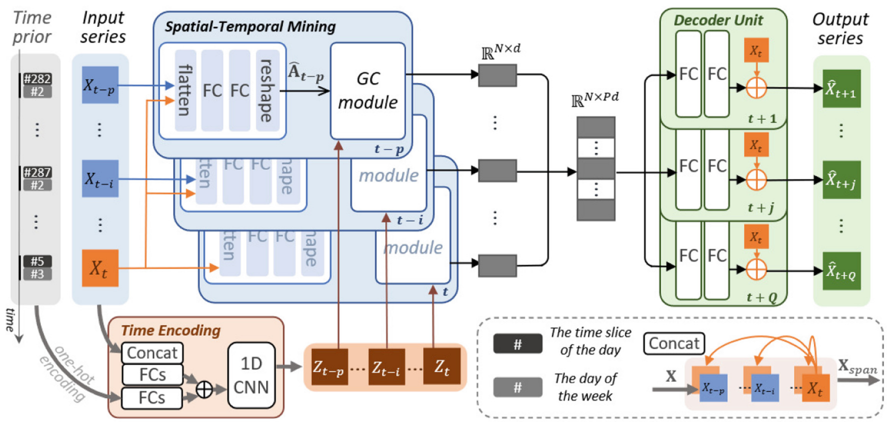

Fig. 1. The overall framework of the proposed TAGnn model.

The overall framework of the proposed TAGnn model is illustrated in Fig. 1. The input series $\mathbf{X} = [X_{t-p}, ..., X_{t-i}, ..., X_t] \in \mathbb{R}^{P \times N \times C}$ is reconstructed by data concatenation and combined with the corresponding time prior information $T_{slice} \in \mathbb{R}^{P \times C_1}$ (the time slice of the day, C_1 is the maximum number of time slices in one day, e.g. $C_1 = 288$ when each time slice is 5 min) and $T_{day} \in \mathbb{R}^{P \times 7}$ (the day of the week) in the *Time Encoding Module*, they are respectively transformed to hidden features and then aggregated. Then the time-adjoint hidden features $\mathbf{Z} = [Z_{t-p}, ..., Z_{t-i}, ..., Z_t] \in \mathbb{R}^{P \times N \times d}$ (d is the number of the hidden dimension) is generated through a convolution layer by time dimension. After that, each time slice owns a *Spatial-Temporal Mining (STM) Module*. It uses the input feature X_{t-i} in time slice $t - i$ and X_t in the most recent time slice t (X_t only when $i = 0$) to learn the spatial adjacency matrix $\hat{A}_{t-i}$ and then fed $\hat{A}_{t-i}$ and the time-adjoint hidden feature Z_{t-i} into the *Graph Convolution (GC) module*. Outputs of P STM modules are then merged and input to Q decoder units to generate the final prediction in the *Decoder Module*.

4.1 Time Encoding Module

The unique time properties shown in the periodic changes of traffic flow distinguish the traffic forecasting problem from other time series prediction problems. The adjoint temporal information (e.g. the time slice of the day, the day of the week) helps to locate the evolution stage of traffic flow in an intuitive way and readily available. Thus, we input two tensors T_{slice}, T_{day} indicating the temporal tags along with the traffic data into the deep model. All graph nodes in each time slice share the same temporal information, it can be normalized in the day and week scale. For instance, 00:05 am to 00:10 am on Thursday is the *second* time slice (5 min per slice) of the *fourth* day in the week. These two order numbers are encoded to vectors belonging to $\mathbb{R}^{C_1}$ and $\mathbb{R}^7$ by one-hot coding. For the whole input sequence, the adjoint time data are constructed as $T_{slice} \in \mathbb{R}^{P \times C_1}$ and $T_{day} \in \mathbb{R}^{P \times 7}$. Both time labels are transformed into hidden tensors through individual fully-connected layers respectively:

$$H_{slice} = \text{ReLU}(FC_{slice}(T_{slice})) = \text{ReLU}(T_{slice}W_1 + b_1) \in \mathbb{R}^{P \times d} \tag{1}$$

$$H_{day} = \text{ReLU}(T_{day}W_2 + b_2) \in \mathbb{R}^{P \times d} \tag{2}$$

where FC is the fully-connected layer, ReLU is the activation function, $W_1 \in \mathbb{R}^{C_1 \times d}, W_2 \in \mathbb{R}^{7 \times d}$ and $b_1, b_2 \in \mathbb{R}^d$ are the weights and biases of linear projections. H_{slice}, H_{day} are temporal embedding tensors.

For exploring the across-time influence of traffic propagation, we concatenate the items in each time slice $t-i, i \in \{0, 1, ..., p\}$ and the latest time slice t in input traffic data $\mathbf{X}$ to obtain the across-time data feature $\mathbf{X}_{span} \in \mathbb{R}^{P \times N \times 2C}$. Then this reconstructed data feature is transformed into a close dimension through two fully-connected layers:

$$(\mathbf{X}_{span})_i = (X_{t-i}||X_t) \in \mathbb{R}^{N \times 2C} \tag{3}$$

$$\mathbf{H}_{flow} = FC_{span}(\text{ReLU}(\mathbf{X}_{span}W_3 + b_3)) \in \mathbb{R}^{P \times N \times d} \tag{4}$$

where $||$ is the concatenation, $W_3 \in \mathbb{R}^{2C \times d}, b_3 \in \mathbb{R}^d$ are the parameters of linear projection, and $\mathbf{H}_{flow}$ is the transformed traffic feature with across-time representation. This traffic feature and time embedding tensors are added through a broadcasting mechanism since every traffic node shares the same time prior:

$$\mathbf{H}_{ST} = H_{slice} + H_{day} + \mathbf{H}_{flow} \in \mathbb{R}^{P \times N \times d} \tag{5}$$

here, H_{slice} and H_{day} are replicated along with the node dimension and then added with $\mathbf{H}_{flow}$. And $\mathbf{H}_{ST}$ is the deep representation with time-prior of all nodes. To better describe the local temporal pattern of traffic nodes, a convolution layer along with the time dimension is then carried out to obtain the time-adjoint hidden feature $\mathbf{Z}$ with local time trends:

$$\mathbf{Z} = \Theta_1 \star \mathbf{H}_{ST} + a \in \mathbb{R}^{P \times N \times d} \tag{6}$$

where $\star$ denotes the convolution operation, Θ_1 means the $1 \times k$ temporal kernel, k is the kernel size, and a means the bias. To maintain the original time length, we use the replication padding in the time dimension to operate equal-width convolution, which means the first and the last items in $\mathbf{H}_{ST}$ are replicated for the supplement to reach the target length. Here, $\mathbf{Z}$ can be regarded as a matrix sequence $[Z_{t-p}, ...Z_{t-i}, ..., Z_t]$, each item is one of the inputs to the following unit in each time slice.

4.2 Spatial Temporal Mining Modules

The spatial correlations among traffic nodes are time-varying. For a relatively long term, the spatial scope of the impact will become bigger. This motivates us to construct across-time combinations in input series for extracting the spatial-temporal impacts on future traffic evolution. We formulate the influence degree in each time slice as an adjacency matrix, which can be integrated with the time-hidden feature (described in Sect. 4.1) through a graph convolution layer.

Specifically, the raw input data X_{t-i} from each past time slice $t-i$ is connected with the most recent feature X_t and this combination is then flattened. Here features of every node are investigated individually through the flatten operation and their interaction can be mined by simple fully-connected layers. Thus two FC layers transform the flattened result into an adjacency matrix $\hat{A}_{t-i}$:

$$E_{t-i} = \begin{cases} FC_i^1(\text{flatten}(X_{t-i}||X_t)), & i \in \{1, ..., p\} \\ FC_i'(\text{flatten}(X_t)), & i = 0 \end{cases} \in \mathbb{R}^l \tag{7}$$

$$A'_{t-i} = \tanh(FC_i^2(E_{t-i})) \in \mathbb{R}^{N^2} \tag{8}$$

where $E_{t-i} \in \mathbb{R}^l$ is a hidden embedding with size l, and tanh is the activation function. A'_{t-i} is then reshaped into the learned adjacency matrix $\hat{A}_{t-i} \in \mathbb{R}^{N \times N}$ followed by a dropout layer with the rate ϕ. Each entry in $\hat{A}_{t-i}$ can be regarded as the spatial impact degree of a node in past time slice $t-i$ on the other node in the most recent time slice t.

After that, adjacency matrix $\hat{A}_{t-i}$ is fed into the Graph Convolution (GC) module along with the time-adjoint hidden feature Z_{t-i} in the corresponding time slice. In this work, a GC module consists of a spatial-domain graph convolution layer with the GLU activation and a residual connection from input:

$$GC(X, A) = (AX\theta_1 + \beta_1) \odot \sigma(AX\theta_2 + \beta_2) + X \tag{9}$$

where $X \in \mathbb{R}^{N \times d}$ is the interested feature, $A \in \mathbb{R}^{N \times N}$ is the adjacency matrix, $\odot$ denotes element-wise product, σ means the sigmoid function, and $\theta_1, \theta_2 \in \mathbb{R}^{d \times d}, \beta_1, \beta_2 \in \mathbb{R}^d$ are the parameters of projections. The spatial-temporal feature in each time slice $t-i$ is extracted through individual GC module:

$$H_{t-i} = GC_i(Z_{t-i}, \hat{A}_{t-i}) \in \mathbb{R}^{N \times d} \tag{10}$$

Finally, the outputs of STM modules in all P time slices are merged on feature dimension into $H_o \in \mathbb{R}^{N \times Pd}$.

4.3 Decoder Module

The Decoder Module converts the spatial-temporal features from different space-time scales into the final expected targets. To directly model the future evolution of traffic, we sum the input data X_t on the latest time slice t with the hidden features processed by two fully connected layers:

$$\hat{X}_{t+j} = FC_j^4(\text{ReLU}(FC_j^3(H_o))) + X_t \in \mathbb{R}^{N \times C} \tag{11}$$

The setting of the time residual connection of X_t means the model aims to extract future changes based on the latest spatial-temporal situation.

Finally, the expected prediction can be obtained by concatenating the outputs of all Q target time slices:

$$\hat{\mathbf{Y}} = [\hat{X}_{t+1}, \hat{X}_{t+2}, ..., \hat{X}_{t+Q}] \in \mathbb{R}^{Q \times N \times C} \tag{12}$$

where $\hat{\mathbf{Y}}$ is the final predicted sequence.

5 Experiments

5.1 Datasets and Experiment Settings

To evaluate our model, we conduct experiments on four public datasets [2]: PEMS03, PEMS04, PEMS07, PEMS08. All datasets record highway traffic flow in four districts in California. The raw flow data are aggregated every 5 min, thus C_1 is 288. Consistent with previous studies [2,9], all data sets are divided into training, validation, and test sets in a ratio of 6:2:2. The implementation of the proposed model is under the PyTorch framework[1] on a Linux server with one Intel(R) Xeon(R) Gold 5220 CPU @ 2.20 GHz and one NVIDIA Tesla V100-SXM2 GPU card. We use one-hour historical data to predict the next hour's data for all datasets, which means length P and Q are both 12. The number of feature C is 1 for all datasets. We choose Mean Absolute Error (MAE) as the loss function and use Adam as the optimizer. The batch size is 32, the learning rate is 0.001, and the number of training epochs is 100. We use MAE, Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE) [1] to evaluate the prediction performance. The average results of 12 time slices are presented. We compare TAGnn with 10 classical and up-to-date methods: VAR [15], SVR [8], LSTM [3], DCRNN[5], STGCN [12], STFGNN [4], DSTAGCN [14], FCGAGA [7], Graph Wavenet (GWN) [11], and ASTGNN [2].

5.2 Experiment Results

The performance comparison of TAGnn and baseline methods is shown in Table 1, the best results are shown in **bold**. The proposed model TAGnn achieves better performance than baseline methods for most of cases. As Table 1 shown, compared with the most competitive method ASTGNN, our TAGnn achieves 9.28%, 5.92%, 6.69% improvements in terms of MAE, MAPE, RMSE on PEMS08. Likewise, TAGnn has the best performance of all three metrics on PEMS04 and PEMS07, the improvements are 2.25%, 2.19%, 2.55% and 3.24%, 5.16%, 2.86%, respectively. And TAGnn improves ASTGNN by 3.80% in terms of MAE on PEMS03. Instead of stacking many blocks, TAGnn uses a single and independent graph convolution layer in each time slice and captures the spatial-temporal interactive features in parallel through the recombination of input data.

[1] Our codes are available at https://github.com/zhuoshu/TAGnn.

Table 1. Performance comparison on PEMS datasets.

Models	PEMS03			PEMS04			PEMS07			PEMS08		
	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE	MAE	MAPE (%)	RMSE
VAR	23.65	24.51	38.26	23.75	18.09	36.66	75.63	32.22	115.24	23.46	15.42	36.33
SVR	21.97	21.51	35.29	28.70	19.20	44.56	32.49	14.26	50.22	23.25	14.64	36.16
LSTM	21.33	23.33	35.11	27.14	18.20	41.59	29.98	13.20	45.84	22.20	14.20	34.06
DCRNN	18.18	18.91	30.31	24.70	17.12	38.12	25.30	11.66	38.58	17.86	11.45	27.83
STGCN	17.49	17.15	30.12	22.70	14.59	35.55	25.38	11.08	38.78	18.02	11.40	27.83
STFGNN	16.77	16.30	28.34	19.83	13.02	31.88	22.07	9.21	35.80	16.64	10.60	26.22
DSTAGCN	15.31	14.91	25.30	19.48	12.93	30.98	21.62	9.10	34.87	15.83	10.03	24.70
FCGAGA	15.99	17.44	26.99	19.42	15.12	31.33	21.73	9.11	35.33	15.80	10.73	24.73
GWN	14.79	**14.32**	25.51	19.36	13.31	31.72	21.22	9.07	34.12	15.07	9.51	23.85
ASTGNN	14.78	14.79	**25.00**	18.60	12.36	30.91	20.62	8.86	34.00	15.00	9.50	24.70
TAGnn	**14.22**	14.76	25.04	**18.18**	**12.09**	**30.12**	**19.95**	**8.40**	**33.03**	**13.61**	**8.94**	**23.05**

It may help to avoid the difficulty of interpretation to some extent caused by exceedingly increasing the depth of the model. The explicit use of time-prior to node representation and the direct extraction of interactive spatial-temporal dependencies in different time span enables TAGnn to achieve better forecasting performance.

5.3 Ablation Studies and Efficiency Analysis

Our ablation studies are conducted to further validate the effectiveness of the time encoding module, the time residual connection, and the across-time graph convolution modules in the proposed TAGnn model. We design 3 variants: **(a) TAGnn w/o time prior**: It removes the time prior information of the input data. **(b) TAGnn w/o across-time connection**: It cancels the connection of data in each past time slice and the most recent time slice. **(c) TAGnn w/o time residual connection**: It removes the residual connection from the data in the latest time slice.

From the ablation results in Table 2, we can find: (1) Time prior provides considerable gains in traffic prediction tasks and the explicit use of time auxiliary information is helpful for traffic modeling. (2) It's beneficial to consider across-time influence in global spatial mining which is more in line with the characteristics of traffic propagation. (3) By using the time residual connection to learn the future evolution of traffic flow, the framework of TAGnn well describes the temporal evolution contexts and makes a better prediction.

To investigate the proposed model from the efficiency level, we compare the time consumption and convergence curves of TAGnn with ASTGNN and GraphWavenet due to their compelling performance on accuracy. Table 3 shows the training time of each epoch and the total inference time for the whole test data. Note that the training of ASTGNN has two stages (non-autoregressive one and autoregressive one). Figure 2 illustrates the convergence curves, which contain validation MAE in the training process of three models on one-hour prediction. The results demonstrate that TAGnn presents the lowest time consumption among the three methods and quickly achieves lower validation errors than both baseline methods.

Table 2. Performance comparison of the variants of TAGnn.

Models	PEMS04			PEMS08		
	MAE	MAPE(%)	RMSE	MAE	MAPE(%)	RMSE
GraphWavenet	19.36	13.31	31.72	15.07	9.51	23.85
ASTGNN	18.60	12.36	30.91	15.00	9.50	24.70
TAGnn w/o time prior	18.82	12.53	30.53	15.15	9.64	24.03
w/o across-time connection	18.42	12.26	30.30	13.81	9.12	23.30
w/o time residual connection	18.35	12.53	**29.98**	13.94	9.57	**23.05**
TAGnn	**18.18**	**12.09**	30.12	**13.61**	**8.94**	**23.05**

Table 3. Time consumption on PEMS04 dataset.

Models	PEMS04	
	Training (s/epoch)	Inference (s)
GraphWavenet	21.4	2.0
ASTGNN (stage 1)	84.6	42.5
ASTGNN (stage 2)	189.0	
TAGnn	19.2	1.4

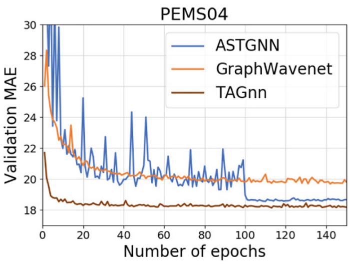

Fig. 2. Model convergence curves on PEMS04 dataset.

6 Conclusion

In this work, we propose a novel Time Adjoint Graph Neural Network (TAGnn) for traffic forecasting. To extract the spatial-temporal interactive features originated from traffic propagation, TAGnn provided a new perspective to process the time series which injects the time prior information into deep representation and connects data of the latest and each past time slices for the following node embedding and spatial modeling. By the data connection across time slices, TAGnn directly capture the long-term and short-term spatial-temporal propagation features. And the introduced time residual connection enables the model to capture the future evolution more easily. The empirical test results validate the accuracy and efficiency of the proposed model. Note the time prior used in this work is plain time identification information independent of the traffic data. When the traffic state changes differently than usual due to events or other factors, the time prior may not necessarily play an auxiliary role. Increasing the robustness of TAGnn against abrupt changes is our future work.

Acknowledgements. This work was supported in part by the National Key Research and Development Program of China under Grant No.2019YFB1704102 and the Fundamental Research Funds for the Central Universities of China under Grant No.2022-4-ZD-05.

References

1. Chicco, D., Warrens, M.J., Jurman, G.: The coefficient of determination r-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput. Sci. **7**, 1–24 (2021)
2. Guo, S., Lin, Y., Wan, H., Li, X., Cong, G.: Learning dynamics and heterogeneity of spatial-temporal graph data for traffic forecasting. IEEE Trans. Knowl. Data Eng. **34**(11), 5415–5428 (2022)
3. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
4. Li, M., Zhu, Z.: Spatial-temporal fusion graph neural networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, no. 5, pp. 4189–4196 (2021)
5. Li, Y., Yu, R., Shahabi, C., Liu, Y.: Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: International Conference on Learning Representations (ICLR) (2018)
6. Liu, D., Wang, J., Shang, S., Han, P.: MSDR: multi-step dependency relation networks for spatial temporal forecasting. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, pp. 1042–1050. KDD 2022, Association for Computing Machinery, New York, NY, USA (2022)
7. Oreshkin, B.N., Amini, A., Coyle, L., Coates, M.: FC-GAGA: fully connected gated graph architecture for spatio-temporal traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 9233–9241 (2021)
8. Ristanoski, G., Liu, W., Bailey, J.: Time series forecasting using distribution enhanced linear regression. In: Pei, J., Tseng, V.S., Cao, L., Motoda, H., Xu, G. (eds.) PAKDD 2013. LNCS (LNAI), vol. 7818, pp. 484–495. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37453-1_40
9. Song, C., Lin, Y., Guo, S., Wan, H.: Spatial-temporal synchronous graph convolutional networks: a new framework for spatial-temporal network data forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 914–921 (2020)
10. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30. Curran Associates, Inc. (2017)
11. Wu, Z., Pan, S., Long, G., Jiang, J., Zhang, C.: Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19, pp. 1907–1913 (2019)
12. Yu, B., Yin, H., Zhu, Z.: Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting. In: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pp. 3634–3640 (2018)
13. Zhang, Q., Chang, J., Meng, G., Xiang, S., Pan, C.: Spatio-temporal graph structure learning for traffic forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 1177–1185 (2020)

14. Zheng, Q., Zhang, Y.: DSTAGCN: dynamic spatial-temporal adjacent graph convolutional network for traffic forecasting. IEEE Trans. Big Data **9**(1), 241–253 (2023)
15. Zivot, E., Wang, J.: Vector autoregressive models for multivariate time series. Model. Financ. Time Series with S-PLUS®, 385–429 (2006)