Journal of Circuits, Systems, and Computers
Vol. 31, No. 6 (2022) 2250112 (17 pages)

DOI: 10.1142/S0218126622501122

A Spatial–Temporal Similar Graph Attention Network for Cyber Physical System Perception via Traffic Forecasting*

Kaidi Zhao†, Mingyue Xu‡, Zhengzhuang Yang§ and Dingding Han¶

Fudan University, Shanghai, P. R. China
†*19110720083@fudan.edu.cn*
‡*19210720131@fudan.edu.cn*
§*19210720144@fudan.edu.cn*
¶*ddhan@fudan.edu.cn*

Received 21 July 2021
Accepted 27 October 2021
Published 15 December 2021

Traffic flow forecasting is the basic challenge in intelligent transportation system (ITS). The key problem is to improve the accuracy of model and capture the dynamic temporal and nonlinear spatial dependence. Using real data is one of the ways to improve the spatial–temporal correlation modeling accuracy. However, real traffic flow data are not strictly periodic because of some random factors, which may lead to some deviations. This study focuses on capturing and modeling the temporal perturbation in real periodic data and we propose a spatial–temporal similar graph attention network (STSGAN) to address this problem. In STSGAN, the spatial–temporal graph convolution module is to capture local spatial–temporal relationship in traffic data, and the periodic similar attention module is to treat the nonlinear traffic flow information. Experiments on three datasets demonstrate that our model is best among all methods.

Keywords: Intelligent transportation system; traffic flows prediction; graph convolution network; attention mechanism.

1. Introduction

Transportation CPS (TCPS) integrates physical objects and cyber systems of traffic. Based on computing, communication and control technologies, transportation information sources and traffic physics are deeply integrated. Through interaction and feedback between information systems and physical systems, transportation CPS realizes the optimization of perception, communication, synergy and decision-making of the traffic system. In the application of intelligent transportation, including intelligent prediction, these applications are all based on single and relatively small independent computing systems. In the future, there is still much work to be done in

*This paper was recommended by Regional Editor Tongquan Wei.
¶Corresponding author.

the in-depth application of collaborative large-scale systems. Edge computing-assisted CPS optimization has been widely discussed in academia and industry.[1–3]

The continuous growth of cars and population has led to a sharp increase in the pressure of urban traffic handling capacity, and many urban traffic are facing severe congestion problems.[4,5] For optimizing the city transportation, many countries try to develop the ITS.[6] Traffic forecasting is an important part of ITS. Accurate forecasting of traffic flow is the benefit to deal with traffic management such as flow control, path navigation and accident handling, which is of importance to enhance traffic efficiency.[7]

Spatio-temporal data can be applied not only to medical care and recommendation systems, but also to transportation.[8–10] Traffic forecasting is to predict the subsequent flow by comprehensively analyzing the spatio-temporal information in traffic data. The key problem is to improve the accuracy of temporal and spatial information. In recent years, deep learning has gradually replaced the application of traditional machine learning in spatial–temporal prediction.[11–15] Convolution neural networks (CNNs),[16–19] recurrent neural networks (RNNs)[20–22] and graph convolution networks (GCNs)[23,24] have complete theoretical applications[1,25,26] and show their great ability in dealing with spatial data and temporal data.

With the development of researches on graphs, graph convolution has been widely used in traffic forecasting. However, due to interference in traffic data, there is still a lack of effective methods to extract spatial and temporal correlation and capture temporal perturbation in periodic data.

Figure 1 shows the dynamic spatial–temporal correlations of traffic congestion at two adjacent time steps. The red line represents that the road is block and the yellow line represents that the road is clear. Due to the upstream and downstream relationship between roads, it can be found that the congestion of A leads to the congestion of B and C in the next time step. This indicates that the traffic between adjacent roads has a strong correlation in adjacent time steps. Besides, it can also be found that the impact of A on different neighbors is different. Previous studies[27] capture the dynamic of data by stacking spatial–temporal convolution blocks. They only consider the spatial correlation under the same temporal dimension but not the vertex domain span time dimension. There is a limitation that many existing studies

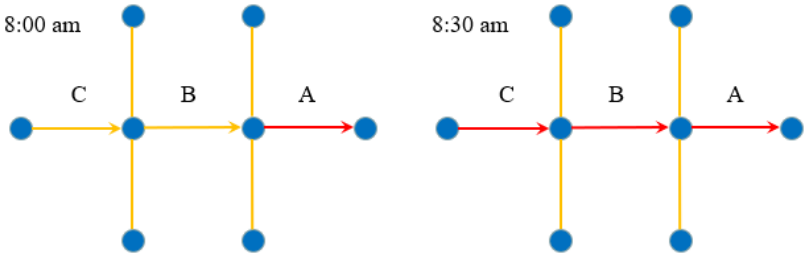

Fig. 1. The spatial–temporal correlation of traffic flow.

ignore the temporal perturbation in periodic traffic data. Due to many random factors such as traffic accidents and traffic jams, daily traffic trends are similar but not identical. For example, the peak hour may be at 8:30 a.m. on Monday but at 9:00 a.m. on Tuesday. Though previous studies[27] consider periodicity, they fail to consider the temporal perturbation in periodic data.

To treat the problems mentioned above, we set a graph framework named spatial–temporal similar graph attention network (STSGAN). First, considering the spatial–temporal correlation, we construct spatial–temporal structure by connecting neighbor graphs in adjacent time steps into one graph as input to the model. Then we construct Spatio–Temporal Graph Convolution Module (STGC), which can model the spatiao-temporal correlations in these localized spatial–temporal graphs. We deploy short-term encoder and long-term encoder to extract current and historical spatial–temporal correlations. In order to learn the similarity in periodic historical data, we design a periodic similar attention module to tackle the limitation of temporal perturbation in periodic traffic data. Finally, in the output, the attention mechanism is used to weigh the different historical time steps on this kind of forecasting.

The contributions are given as follows:

- We constructed a novel dynamic graph neural networks framework named STSGAN to predict traffic flow. We design a spatial–temporal graph network and set a spatial–temporal graph convolution module based on GCN to capture correlations in the spatial–temporal graph structure.
- We construct a periodically similar attention module to model the similarity in periodic historical data; it learns the different importance of nodes in adjacent time steps by tackling the limitation of temporal perturbation.
- We conduct the experiments with some baseline methods on three datasets. This method can effectively help the model to improve accuracy.

The remainder of this paper is structured as follows. We review the related works in Sec. 2. Then the traffic preliminaries are introduced in Sec. 3. In Sec. 4, we discuss the components of the STSGAN model. We experiment with our model in Sec. 5. A conclusion is given in Sec. 6.

2. Related Work

2.1. *Traffic forecasting*

Traffic forecasting has been widely concerned for a long time. As reported by Li,[28] the major models for traffic forecasting are machine learning models. The former includes ARIMA model[29] and Kalman filter.[30] It is not consistent with the nonlinearity of urban traffic dynamics. Moreover, these models are usually aimed at specific small areas, but cannot handle city-level traffic flow data. For example, ARIMA needs

uninterrupted time series data and performs not well with incomplete data. Compared with classical model such as support vector machine model,[31] K-nearest neighbor model,[32] Bayesian networks,[33] deep learning models which use Recurrent Neural Network (RNN) and CNNs have shown better performance in dealing with traffic data.[34–36] Hybrid models that extract the spatio-temporal dependency by stacking CNN and Long Short-Term Memory (LSTM) modules also perform well.[37–39] Some researchers show that festivals and weather factors can also affect the accuracy of the model.[40,41]

Deep learning models perform well with traffic problems because they can explore complex spatial–temporal features. Some methods learn the hierarchical features of grid traffic data using CNNs.[14,16,34] To model the periodic data in traffic flow, some methods use RNNs to extract the temporal features.[22,28] Graph convolution has gradually become the mainstream method of spatial–temporal correlation modeling, and an increasing number of studies have attempted to apply graph convolutions to learn different levels of features for traffic forecasting. Yu *et al.*[42] proposed a gated graph convolution module for traffic prediction but did not consider the dynamic spatio-temporal dependence of traffic data. Li *et al.*[28] combined diffusion convolutions[43] to capture the spatial correlation by an encoder–decoder architecture with scheduled sampling. But, this method does not show a strong temporal correlation. Wu *et al.*[44] proposed a model with deep spatial–temporal dependence that can learn the hidden spatial information through node embedding and using a stack dilated 1-Dimension convolution to expand the receptive field exponentially. This method improves the extensibility of the model, but some details are missing in the modeling of local spatio-temporal features. In addition, Bai *et al.*[45] stacked the gated residual GCN modules and attention mechanisms to model spatial–temporal correlations simultaneously. Compared to the proposed STSGAN model, however, the spatio-temporal correlations of each node in different time steps are ignored in its concatenate model. The above model ignores the impact of data in different time steps in their current state. All of these models used independent components to capture spatio-temporal dependencies.

2.2. *Graph convolution network*

Graph convolution network has achieved an excellent performance in dealing with non-Euclidean data. Two mainstream approaches to graph convolution are spatial and spectral methods. A spectral method proposed by Bruna[46] is a general graph convolution framework. Defferrard *et al.*[47] reduced the computational complexity of the general graph convolution framework using Chebyshev polynomial approximation. In this algorithm, spectral decomposition of the Laplace matrix is avoided but the time complexity is very high. Kipf[48] simplified ChebNet and achieved the best performance. GCN regards the network as a graph and aggregates the information of adjacent nodes into features. The coefficients are calculated by spatial information

(for example, the distance between nodes). In addition, space convolution operation extends the traditional convolution network from Euclidean space to the vertical domain. It avoids multiplying matrices multiple times. However, excessive smoothing may occur when the number of network layers increases, and the features of each node in the network are similar and difficult to distinguish. Hamilton *et al.*[49] gave a method that the embedding vector of the target vertex is generated by learning a function of an aggregate representation of the neighbor vertices. Velickovic *et al.*[50] proposed a graph attention network with attention layers to adjust the importance of the neighbor nodes. The node aggregation ability is improved effectively.

3. Spatial–Temporal Similar Graph Attention Network

This study defines the traffic network as a weighted graph $G = (V, E, A)$ where V is a set of $|V| = N$ nodes, E denotes a set of edges, $A \in \mathbb{R}^{N\times N}$ denotes the weighted adjacency matrix of graph G and each element in $A_{i,j}$ means the distance between nodes i and j. The undirected graph G is the spatial dependence of nodes in the dimension.

The graph signal $X_t \in \mathbb{R}^{N\times C}$ is the observation of the network G at time t. C counts the node features. According to the observations of the historical measurements over the previous P time steps $\chi = (X_1, X_2, \ldots, X_P)$, the target is to predict the subsequent results in next Q steps in $\hat{Y} = (\hat{X}_{P+1}, \hat{X}_{P+2}, \ldots, \hat{X}_{P+Q})$ for all networks.

Figure 2 is our STSGAN model. It is divided into long- and short-term components. The former contains an STGC module and a periodically similar attention module. The short-term component contains only an STGC module. To construct a partial spatial–temporal module, every node is connected to itself in the previous adjacent time steps first. We then divide the data into daily and hourly time frames near the target time steps. These two data fragments use long-term and short-term components to model the corresponding spatio-temporal correlations, respectively. The STGC module is applied to study the dependence in localized spatio-temporal graph. The periodically similar attention module is used to tackle the limitation in which the traffic information is not periodic in the daily data. Finally, the output module with an attention mechanism is used to fuse long- and short-term representations. The model is introduced below.

3.1. *Localized spatial–temporal graph*

As a limitation to the temporal correlation of the previous methods, the nodes are only connected with themselves at the adjacent time steps, and not their neighbors. Topological structure plays a key role in node representation. Inspired by Li's approach,[51] which demonstrated the importance of neighbors belonging to adjacent time steps, we design a localized spatial–temporal graph that connects the nodes to

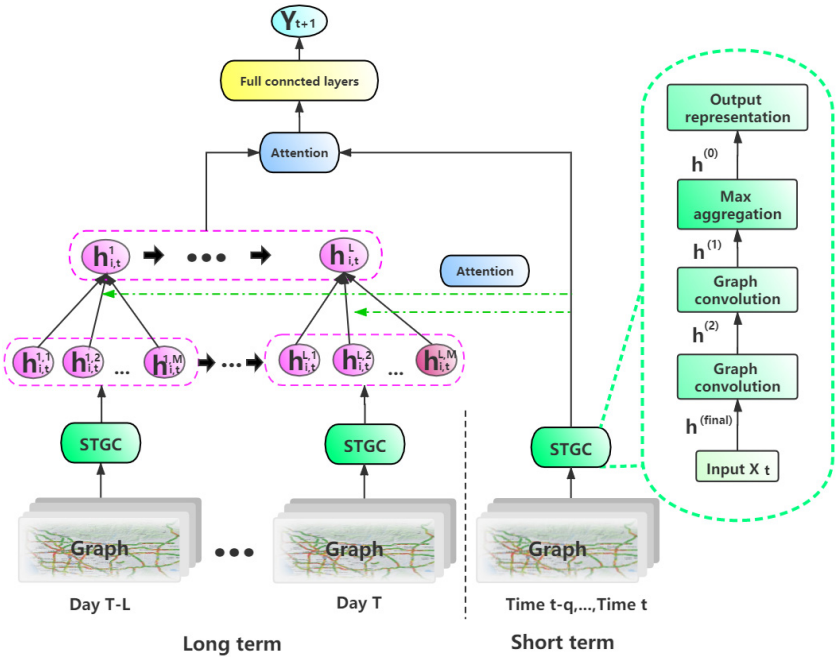

Fig. 2. The architecture of STSGAN.

all neighbors at the current and adjacent time steps (Fig. 3). The black solid lines denote the correlations between a node and its neighbors at the current time steps, and the green dashed lines denote the correlations between a node and its neighbors at the adjacent time steps. By connecting all nodes with their neighbors at the current and adjacent time steps, we can directly learn the dynamic spatial dependence and temporal dependence of the traffic network simultaneously.

$A \in \mathbb{R}^{N\times N}$ is the adjacency matrix of the spatial graph. The localized spatio-temporal graph on two continuous spatial graphs is $A' \in \mathbb{R}^{2N\times 2N}$. If two nodes

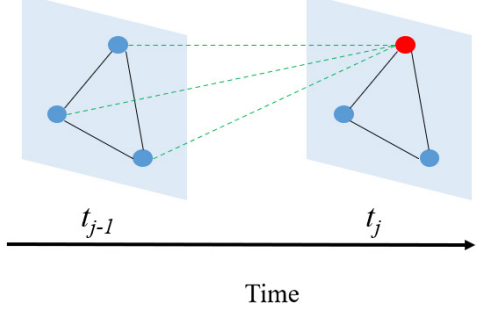

Fig. 3. Localized Spatial–Temporal Graph.

connect with each other in this localized spatial–temporal graph, the value in A' is set to 1. The adjacency matrix can be obtained as follows:

$$A'_{i,j} = \begin{cases} 1, & i \text{ connect to } j, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

3.2. *Long- and short-term encoders*

Temporal networks exhibit multiple system states, and the detection of different time states in the temporal network data help reveal the underlying dynamic process of the network.[52] Many previous studies have considered only one-step forecasting, in which the traffic only flows during the next time step. These models only reduce the error for next time-step forecasting without considering the subsequent time steps, which leads to poor performance for multi-step forecasting. Most previous studies on temporal correlations have used encoder–decoder architectures based on LSTM as the encoders and decoders.[51] In addition, there are few studies that have considered the temporal correlations by GCN.

In this study, we use an STGC to extract the spatio-temporal dependency in localized spatio-temporal graph mentioned above. In the long-term encoder, we take the previous L days of data and for each previous day, we take p time steps adjacent to the target time step and then apply the localized spatial–temporal graph on the traffic flow sequences $(X_L^1, X_L^2, \ldots, X_L^p)$ to obtain the new spatial–temporal graph sequence $(X_L'^1, X_L'^2, \ldots, X_L'^{p-1})$ as an input to capture the historical spatial–temporal correlations. Compared with the model that uses RNN to capture the time correlation, GCN decreases the number of iterations, thereby effectively reducing the loss of information. We then feed the historical spatial–temporal representation close to the current time into the periodically similar attention module to obtain a daily representation. The output sequence of the long-term component is Y_L.

The short-term encoder is used to predict the current traffic flow. It takes most recent q time steps to capture the short-term spatial–temporal correlations. We then combine the localized spatio-temporal graph structure and the STGC to capture the spatial–temporal dependence. Except for periodically similar attention module in long-term encoder, the short-term encoder takes the same operations. We apply the localized spatial–temporal graph on the short-term traffic flow sequences $(X_t^1, X_t^2, \ldots, X_t^q)$ to obtain the new short-term sequence $(X_t'^1, X_t'^2, \ldots, X_t'^{q-1})$ as an input. The output of the short-term component is Y_s.

3.3. *Spatial–temporal graph convolution module*

We design the STGC module to construct the relationship between temporal and spatial information in the localized spatial–temporal graph structure mentioned above. First, we extract the adjacency matrices of two adjacent time steps from the

localized spatial–temporal graph A' and then use graph convolution to obtain the local spatial–temporal representation. We then deploy a full layer to transform the features of the nodes as the new input:

$$\mathrm{GCN}(h^{(l-1)}) = h^{(l)} = \sigma(A'h^{(l-1)}W + b) \in \mathbb{R}^{2N\times C}. \tag{2}$$

$A' \in \mathbb{R}^{2N\times 2N}$ is adjacency matrix of localized spatial–temporal graph, $h^{(l-1)} \in \mathbb{R}^{2N\times C}$ denotes the input of the l convolution layer and $W \in \mathbb{R}^{C\times C'}$ and $b \in \mathbb{R}^{C'}$ are learnable parameters. We select the gated linear units (GLU)[53] after graph convolution. The GLU can control the information and send it to the next layer.

$$h^{(l)} = \sigma(A'h^{(l-1)}W_1 + b_1) \otimes \mathrm{sigmoid}(A'h^{(l-1)}W_2 + b_2), \tag{3}$$

where $W_1, W_2 \in \mathbb{R}^{C\times C'}$ and $b_1, b_2 \in \mathbb{R}^{C'}$ are learnable parameters with sigmoid activation function; $\otimes$ is element-wise product.

We stack multiple graph convolution layers to increase receptive field, allowing it to aggregate the features of each node with its direct and high-order neighbors. We select JK-net[54] as the base structure of our STSGAN. As shown in Fig. 2, which details the STGC, $h^{(l)}$ is output of the l graph convolution. $h^{(0)}$ is input of STGC. The output of each graph convolution layer $h^{(l)}$ will be aggregated and compacted. We select max-pooling as the aggregation operation applied to the outputs of all graph convolutions. The max operation is formulated as follows:

$$h_{\max} = \max(h^{(1)}, h^{(2)}, \ldots, h^{(\mathrm{final})}) \in \mathbb{R}^{2N\times C}. \tag{4}$$

3.4. *Periodically similar attention module*

In this part, we model the periodic dependency in the long-term encoder. We deploy periodically similar attention module to capture the temporal perturbation in periodic data.

We use STGC module to model the periodic dependency. Existing methods only consider the same time segment in historical data to extract the periodic information. Differing from all previous studies, we consider the temporal perturbation in temporal correlations. Traffic data are not strictly periodical, and data from the same moment can deviate from day to day. Temporal perturbations of periodic information are ubiquitous in traffic data. For example, there always is a morning rush hour on weekdays but the time can vary from 8:30 p.m. to 11:30 p.m. Figure 4 shows the intensity of traffic flow and time in a week. We set the max traffic flow in a week as 1.0. The rest is the ratio to it. The curve is obtained after the overall flow is smoothed, and the trend of the overall flow can be seen. Although traffic series are periodic in terms of their trend, the actual peaks of such series occur at completely different times of the day. Thus, we design a periodic similar attention module to get the temporal perturbation. It is mainly weighted the different effects in daily periodicity. For all previous days L, we select M time intervals as reference segments for

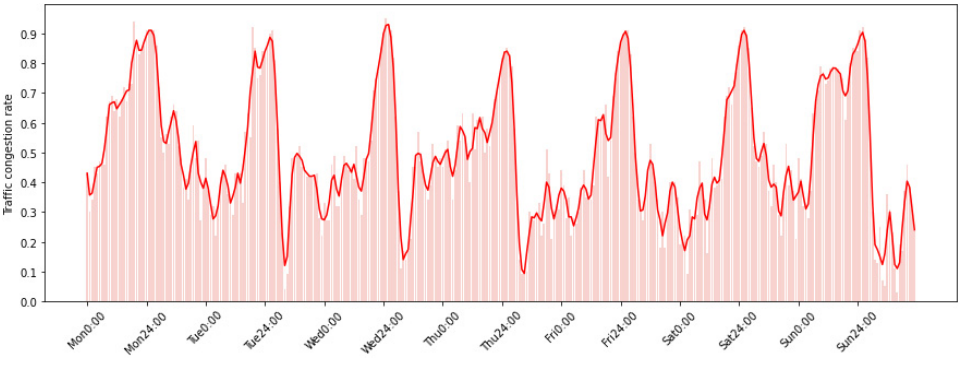

Fig. 4. Temporal perturbation of periodic data between different days of the week.

the target time step. If the target time is 8:30–9:00 a.m., we choose M time intervals before and after the target time (7:30–10:00, where $M = 4$). These time intervals $m \in M$ indicate the temporal perturbation. We use the STGC module to model $l \in L$ day temporal information; it is formulated as follows:

$$h_{i,t}^{l,m} = \mathrm{STGC}(Y_{i,t}^{l,m}, h_{i,t}^{l,m-1}), \tag{5}$$

where $h_{i,t}^{l,m}$ is the previous l days for the predicted time t of node i. In addition, $Y_{i,t}^{l,m}$ denotes the input of STGC at time m during the previous day l with time t of node i.

We deploy attention module to weigh the related segments during each previous day. The representation of $h_{i,t}^{l}$ for each previous day is a weighted sum of each time segment m and is defined as follows:

$$h_{i,t}^{l} = \sum_{m \in M} \alpha_{i,t}^{l,m} h_{i,t}^{l,m}, \tag{6}$$

where $\alpha_{i,t}^{l,m}$ is the similarity between the time segment m and the target time interval during day $l \in L$, which is derived by comparing the learned spatial–temporal representation from the STGC module. In addition, $\alpha_{i,t}^{l,m}$ is defined as follows:

$$\alpha_{i,t}^{l,m} = \frac{\exp(\mathrm{score}(h_{i,t}^{l,m}, h_{i,t}))}{\sum_{m \in M} \exp(\mathrm{score}(h_{i,t}^{l,m}, h_{i,t}))}. \tag{7}$$

The score function is regarded as a content-based function, which is defined in the following manner:

$$\mathrm{score}(h_{i,t}^{l,m}, h_{i,t}) = v^T \tanh(W_H h_{i,t}^{l,m} + W_X h_{i,t} + b_X), \tag{8}$$

where W_H, W_X, b_X and v are learned parameters.

Finally, we obtain a periodic representation $h_{i,t}^{l}$ during each previous day $Y_L = (h_{i,t}^{1}, h_{i,t}^{2}, \ldots, h_{i,t}^{L})$.

3.5. *Attention-based output module*

We fuse the long-term representation using the attention mechanism to weigh the previous daily sequence Y_L as follows:

$$h'_{i,t} = \sum_{l \in L} \alpha_{i,t}^{l} h_{i,t}^{l}, \tag{9}$$

where $\alpha_{i,t}^{l}$ is the weight of each previous day and is defined as follows:

$$a_{i,t}^{l} = \frac{\exp(\text{score}(h_{i,t}^{l}, h_{i,t}))}{\sum_{l \in L} \exp(\text{score}(h_{i,t}^{l}, h_{i,t}))}. \tag{10}$$

The long-term representation is transformed into $Y'_L = (h_{i,t}^{\prime 1}, h_{i,t}^{\prime 2}, \ldots, h_{i,t}^{\prime L})$. Here, we directly merge the long- and short-term representations as follows:

$$Y_{L+s} = \tanh(Y_L^{'} + Y_s) \tag{11}$$

Tanh is a tangent to scale the values between -1 and 1. We choose a fully connected layer to transform Y_{l+s} into expected prediction as follows:

$$\hat{Y} = \text{ReLU}(Y_{L+s} W_1 + b_1) \cdot W_2 + b_2, \tag{12}$$

where W_1, W_2, b_1 and b_2 are learnable parameters.

4. Experiments

4.1. *Datasets*

We validate the model on PeMSD4, PeMSD7, and PeMSD8.[55] The data are aggregated into 5-min intervals. All detectors are deployed along the highway in California. We can obtain geographic information from these datasets.

The weighted adjacency matrix in the above datasets is calculated by the distances among road sensor stations. According to STGCN,[42] the weighted adjacency matrix W is

$$w_{ij} = \begin{cases} \exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right), & i \neq j \text{ and } \exp\left(-\dfrac{d_{ij}^2}{\sigma^2}\right) \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

where w_{ij} is the weight of edge. Both σ^2 and ϵ are variable of the matrix W. The data on the first 80% days are the train set and the remaining are test set.

Table 1. Dataset description.

Datasets	Sensors	Time range
PeMSD04	307	1/1/2018 to 2/28/2018
PeMSD07	883	5/1/2017 to 8/31/2017
PeMSD08	170	7/1/2016 to 8/31/2016

4.2. *Preprocessing*

The missing values are filled in according to linear interpolation. The detailed data are shown in Table 1. All data are transformed through zero-mean normalization using the mean to standardize the data:

$$x' = \frac{x - \text{mean}(X)}{\text{std}(X)}. \tag{14}$$

4.3. *Setting*

We implement the STSGAN model based on the TensorFlow framework. According to the study by Kipf and Welling,[48] we tested the Chebyshev polynomial $K = 1$. The MSE and RMSE indexes are used as loss function. The batch size is 64 and the learning rate is 0.0001.

4.4. *Comparison and analysis of results*

We compare our model with the following baselines:

- GRU[56]: Gated recurrent unit network.
- DCRNN[28]: Diffusion convolution RNN utilizing diffusion GCNs and seq2seq to encode two kinds of data, respectively.
- STGCN[42]: Spatial–temporal GCN, which uses ChebNet and 2D convolution networks to learn spatio–temporal correlations.
- LSTM[57]: Long Short-Term Memory network.
- ASTGCN[27]: Attention-based spatial–temporal GCN with spatial attention and temporal attention mechanisms.
- STG2Seq[45]: Spatio-temporal graph-to-sequence model using a multi-gated graph convolution module and the seq2seq architecture with attention mechanisms to achieve multi-step forecasting.

We compare the STSGAN with the above baselines on the PeMSD4, PeMSD7 and PeMSD8 datasets. Table 2 shows the results of each methods.

Table 2 shows our STSGAN significantly outperforms all baseline for all evaluation metrics. STSGAN achieves the lowest RMSE of 33.14 on the PeMSD4 dataset, and 38.49 and 25.41 on the PeMSD7 and PeMSD8 datasets, and it also achieves the

Table 2. Performance comparison of different approaches for traffic flow forecasting.

	PeMSD4		PeMSD7		PeMSD8	
Model	RMSE	MAE	RMSE	MAE	RMSE	MAE
LSTM	45.82	29.45	44.97	30.13	36.96	23.18
GRU	45.11	28.65	45.17	30.08	35.95	22.20
DCRNN	38.12	24.70	37.61	25.08	27.83	18.26
STGCN	38.29	25.15	38.53	25.91	27.87	18.88
ASTGCN	33.49	22.93	42.88	29.17	26.33	17.69
STG2Seq	36.27	23.75	47.33	33.10	27.71	18.17
STSGAN	**33.02**	**20.74**	**38.49**	**24.86**	**25.31**	**16.43**

lowest MAE of 20.54 on the PeMSD4 dataset, and 24.86 and 16.50 on the PeMSD7 and PeMSD8 datasets, respectively. We can see from Table 2 that LSTM and GRU, which cannot utilize the spatial dependencies, do not perform well, and only take temporal correlations into consideration. It is hard to model the traffic data effectively when depending only on the temporal information. The DCRNN, STGCN, ASTGCN, STG2Seq, and our proposed STSGAN models all take spatial information based on temporal information, and thus, they achieve a better performance than the methods using only time-series forecasting.

The DCRNN, STGCN, and ASTGCN models use two modules to model the spatial dependencies, one module for extracting the spatial correlation when considering only the current time step, which separate the relationship between temporal and spatial information. Our method applies a localized spatial–temporal graph structure, and the results in Table 2 indicate that STSGAN is better than these methods, demonstrating the advantage of a localized spatial–temporal graph structure. As we can see from Table 2, both ASTGCN and STG2Seq perform better than STGCN and DCRNN, thereby indicating the importance of historical information. As a limitation of ASTGCN and STG2Seq, they simply concatenate on the features of the long-term information, rather than modeling the perturbations in such information. Our STSGAN considers the local spatial–temporal correlation and captures the time perturbation in the spatial–temporal data.

4.5. *Component analysis*

To study the impact of different modules in STSGAN, we design five different STSGAN models. Different parts of each model were cut to study the impact of this part. We compared the five variants on the PeMSD4. The differences in these five models are described as follows:

- Basic: This model does not have an STGC or periodically similar attention module. Rectified Linear Unit is the activation function of model. The convolution layer contains 12 filters to generate the forecasting.

Table 3. Component analysis of STSGAN.

	RMSE	MAE
Basic	35.46	23.39
Spatial–temporal graph convolution	34.82	21.67
GLU	34.53	21.41
Mask	34.26	20.97
Temporal perturbation	33.02	20.74

- Spatial–temporal graph convolution: This model applies the STGC in the basic model.
- GLU: This model changes all activation functions in the multi-module model into the GLU.
- Mask: We use a learnable mask matrix to adjust the weights in the adjacency matrix. This model is based on GLU version, the mask version.
- Temporal perturbation: This model adds a periodically similar attention module based on the mask version.

The model applies the STGC during every period and the result outperforms the basic model, which shows the necessity of modeling the spatial–temporal correlation in the traffic data. As illustrated in Table 3, GLU achieves a better performance than ReLU because GLU has twice the parameter size, and thus, its larger capacity enables it to capture complex spatial–temporal correlations. In addition, it can control the output more flexibly than ReLU.

The results of the temporal perturbation show that extracting the temporal perturbation information in periodic data can clearly improve the performance. We add the mask matrix to tune the weights during graph convolution operations. The results show that it has significant influence on performance.

5. Conclusion

We propose a graph framework called STSGAN for traffic forecasting in this paper. This framework is based on spatial–temporal graphs and uses STGC to capture spatial–temporal correlations. It also contains a periodically similar attention module to tackle the limitation in which traffic information in daily data is not strictly periodic. Moreover, according to the experimental result, the proposed model performs better than other baselines.

In addition to the time perturbation in the periodic data mentioned in this paper, there are many other factors that affect the accuracy. In a future study, we will consider more factors on traffic flow forecasting as the STSGAN can be regarded as a general framework for forecasting spatial–temporal data to be applied to other tasks.

Acknowledgments

This study was supported by the National Key R&D Program of China under Grant No. 2018YFB2101302. We also acknowledge the NSF awards 11875133

and 11075057. I gratefully acknowledge the help of Xihou Zhang, who provided Fig. 2.

References

1. Y. Zhang, Y. Li, X. Zhang and S. Zheng, Prediction method of Nox from power station boilers based on neural network, *J. Circuits Syst. Comput.* **30** (2021) 2150097.
2. K. Cao, S. Hu, Y. Shi, A. Colombo, S. Karnouskos and X. Li, A survey on edge and edge-cloud computing assisted cyber-physical systems, *IEEE Trans. Ind. Inform.* **17**(11) (2021) 7806–7819.
3. K. Cao, Y. Cui, Z. Liu, W. Tan and J. Weng, Edge intelligent joint optimization for lifetime and latency in large-scale cyber-physical systems, *IEEE Internet Things J.* **16** (2021) 12716–12619.
4. S. Sun, R. Huang and Y. Gao, Network-scale traffic modeling and forecasting with graphical lasso and neural networks, *J. Transp. Eng.* **138** (2012) 1358–1367.
5. C. Xu, A novel recommendation method based on social network using matrix factorization technique, *Inform. Process. Manage.* **54** (2018) 463–474.
6. J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu and C. Chen, Data-driven intelligent transportation systems: A survey, *IEEE Trans. Intell. Transp. Syst.* **12** (2011) 1624–1639.
7. J. Yuan, Y. Zheng, X. Xie and G. Sun, Driving with knowledge from the physical world, in *Proc. 17th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining*, San Diego, California, USA, August 2011, pp. 316–324.
8. R. Kunz and R. Tetzlaf, Spatio-temporal dynamics of brain electrical activity in epilepsy: Analysis with cellular neural networks (CNNs), *J. Circuits Syst. Comput.* **12** (2003) 825–844.
9. C. Xu, A. S. Ding and K. Zhao, A novel POI recommendation method based on trust relationship and spatial–temporal factors, *Electron. Commer. Res. Appl.* **48** (2021) 101060.
10. K. Xiang, C. Xu and J. Wang, Understanding the relationship between tourists' consumption behavior and their consumption substitution willingness under unusual environment, *Psychol. Res. Behav. Manag.* **14** (2021) 483.
11. R. Kunz and R. Tetzlaff, Spatio-temporal dynamics of brain electrical activity in epilepsy: Analysis with cellular neural networks (CNNs), *J. Circuits Syst. Comput.* **12** (2003) 825–844.
12. Y. Lv, Y. Duan, W. Kang, Z. Li and F.-Y. Wang, Traffic flow prediction with big data: A deep learning approach, *IEEE Trans. Intell. Transp. Syst.* **16** (2014) 865–873.
13. J. Zhang, Y. Zheng, D. Qi, R. Li, X. Yi and T. Li, Predicting citywide crowd flows using deep spatio-temporal residual networks, *Artif. Intell.* **259** (2018) 147–166.
14. N. G. Polson and V. O. Sokolov, Deep learning for short-term traffic flow prediction, *Transp. Res. Part C Emerg. Technol.* **79** (2017) 1–17.
15. S. Sengar and X. Liu, An efficient load forecasting in predictive control strategy using hybrid neural network, *J. Circuits Syst. Comput.* **29** (2020) 567–576.
16. 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* **17** (2017) 818.
17. D. C. Park, Long-term prediction of ethernet traffic using multiscale-bilinear recurrent neural network with adaptive learning, *J. Circuits Syst. Comput.* **19** (2010) 155–171.

18. M. Fouladgar, M. Parchami, R. Elmasri and A. Ghaderi, Scalable deep traffic flow neural networks for urban traffic congestion prediction, in *2017 Int. Joint Conf. Neural Networks (IJCNN)* (IEEE, 2017), pp. 2251–2258.
19. B. Du, H. Peng, S. Wang, M. Z. A. Bhuiyan, L. Wang, Q. Gong, L. Liu and J. Li, Deep irregular convolutional residual LSTM for urban traffic passenger flows prediction, *IEEE Trans. Intell. Transp. Syst.* **21** (2019) 972–985.
20. Y. Tian and L. Pan, Predicting short-term traffic flow by long short-term memory recurrent neural network, in *2015 IEEE Int. Conf. Smart City/SocialCom/SustainCom (SmartCity)* (IEEE, 2015), pp. 153–158.
21. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen and J. Liu, LSTM network: A deep learning approach for short-term traffic forecast, *IET Intell. Transp. Syst.* **11** (2017) 68–75.
22. Z. Zhene, P. Hao, L. Lin, X. Guixi, B. Du, M. Z. A. Bhuiyan, Y. Long and D. Li, Deep convolutional mesh RNN for urban traffic passenger flows prediction, in *2018 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (IEEE, 2018), pp. 1305–1310.
23. Q. Zhang, Q. Jin, J. Chang, S. Xiang and C. Pan, Kernel-weighted graph convolutional network: A deep learning approach for traffic forecasting, in *2018 24th Int. Conf. Pattern Recognition (ICPR)* (IEEE, 2018), pp. 1018–1023.
24. M. Wang, B. Lai, Z. Jin, Y. Lin, X. Gong, J. Huang and X. Hua, Dynamic spatio-temporal graph-based CNNs for traffic prediction, preprint (2018) arXiv:1812.02019.
25. D.-C. Park, Long-term prediction of ethernet traffic using multiscale-bilinear recurrent neural network with adaptive learning, *J. Circuits Syst. Comput.* **19** (2010) 155–171.
26. H. R. Jiang and K. S. Kwak, On modified complex recurrent neural network adaptive equalizer, *J. Circuits Syst. Comput.* **11** (2002) 93–101.
27. S. Guo, Y. Lin, N. Feng, C. Song and H. Wan, Attention based spatial-temporal graph convolutional networks for traffic flow forecasting, in *Proc. AAAI Conf. Artificial Intelligence*, Vol. 33 (2019), pp. 922–929.
28. Y. Li, R. Yu, C. Shahabi and Y. Liu, Diffusion convolutional recurrent neural network: Data-driven traffic forecasting, in *6th Int. Conf. Learning Representations (ICLR 2018)* (2018), pp. 1–16.
29. M. S. Ahmed and A. R. Cook, Analysis of freeway traffic time-series data by using Box-Jenkins techniques, *Transp. Res. Rec.* (1979).
30. I. Okutani and Y. J. Stephanedes, Dynamic prediction of traffic volume through Kalman filtering theory, *Transp. Res. B Methodol.* **18** (1984) 1–11.
31. M. Castro-Neto, Y.-S. Jeong, M.-K. Jeong and L. D. Han, Online-SVR for short-term traffic flow prediction under typical and atypical traffic conditions, *Expert Syst. Appl.* **36** (2009) 6164–6173.
32. P. Cai, Y. Wang, G. Lu, P. Chen, C. Ding and J. Sun, A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting, *Transp. Res. Part C Emerg. Technol.* **62** (2016) 21–34.
33. G. Fusco, C. Colombaroni and N. Isaenko, Short-term speed predictions exploiting big data on large urban road networks, *Transp. Res. Part C Emerg. Technol.* **73** (2016) 183–201.
34. J. Zhang, Y. Zheng and D. Qi, Deep spatio-temporal residual networks for citywide crowd flows prediction, in *Thirty-First AAAI Conf. Artificial Intelligence* (AAAI Press, 2017), pp. 1655–1661.

35. 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, in *Thirty-Second AAAI Conf. Artificial Intelligence* (AAAI Press, 2018), pp. 2588–2595.
36. H. Yao, X. Tang, H. Wei, G. Zheng and Z. Li, Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction, in *Proc. AAAI Conf. Artificial Intelligence*, Vol. 33 (2019), pp. 5668–5675.
37. X. Zhou, Y. Shen, Y. Zhu and L. Huang, Predicting multi-step citywide passenger demands using attention-based neural networks, in *Proc. Eleventh ACM International Conf. Web Search and Data Mining* (2018), pp. 736–744.
38. S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong and W.-C. Woo, Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in *Advances in Neural Information Processing Systems* (2015), pp. 802–810.
39. J. Ke, H. Zheng, H. Yang and X. M. Chen, Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach, *Transp. Res. Part C Emerg. Technol.* **85** (2017) 591–608.
40. S. Wang, X. Zhang, J. Cao, L. He, L. Stenneth, P. S. Yu, Z. Li and Z. Huang, Computing urban traffic congestions by incorporating sparse GPS probe data and social media data, *ACM Trans. Inform. Syst.* **35** (2017) 1–30.
41. Z. Lin, J. Feng, Z. Lu, Y. Li and D. Jin, DeepSTN+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis, in *Proc. AAAI Conf. Artificial Intelligence*, Vol. 33 (2019), pp. 1020–1027.
42. B. Yu, H. Yin and Z. Zhu, Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting, in *Proc. 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with the 2017 Int. Joint Conf. Artificial Intelligence* (Stockholm, Sweden, 2017), pp. 3634–3640.
43. J. Atwood and D. Towsley, Diffusion-convolutional neural networks, in *Advances in Neural Information Processing Systems* (2016), pp. 1993–2001.
44. Z. Wu, S. Pan, G. Long, J. Jiang and C. Zhang, Graph WaveNet for deep spatial-temporal graph modeling, in *Proc. 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with the 2019 Int. Joint Conf. Artificial Intelligence* (Macao, China, 2019), pp. 1907–1913.
45. L. Bai *et al.*, STG2Seq: Spatial-temporal graph to sequence model for multi-step passenger demand forecasting, in *Proc. 9th Workshop on Multimedia for Cooking and Eating Activities in Conjunction with the 2019 Int. Joint Conf. Artificial Intelligence* (Macao, China, 2019), pp. 1981–1987.
46. J. Bruna, W. Zaremba, A. Szlam and Y. LeCun, Spectral networks and locally connected networks on graphs, In *2nd Int. Conf Learning Representations, ICLR 2014, Conference Track Proceedings*, eds Y. Bengio and Y. LeCun (Banff, AB, Canada, 2014).
47. M. Defferrard, X. Bresson and P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in *Advances in Neural Information Processing Systems* (2016), pp. 3844–3852.
48. T. N. Kipf and M. Welling, Semi-supervised classification with graph convolutional networks, in *Int. Conf. Learning Representations (ICLR)* (Toulon, France, 2017).
49. W. Hamilton, Z. Ying and J. Leskovec, Inductive representation learning on large graphs, in *Advances in Neural Information Processing Systems* (2017), pp. 1024–1034.
50. P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò and Y. Bengio, Graph attention networks, in *Int. Conf. Learning Representations* (Vancouver, BC, Canada, 2018).

51. B. Li, X. Li, Z. Zhang and F. Wu, Spatio-temporal graph routing for skeleton-based action recognition, in *Proc. AAAI Conf. Artificial Intelligence*, Vol. 33 (2019), pp. 8561–8568.
52. S. Cao and H. Sayama, Detecting dynamic states of temporal networks using connection series tensors, *Complexity* **2020** (2020) 9649310.
53. Y. N. Dauphin, A. Fan, M. Auli and D. Grangier, Language modeling with gated convolutional networks, *J. Mach. Learn. Res.* **70** (2016) 933–941.
54. K. Xu, C. Li, Y. Tian, T. Sonobe, K.-I. Kawarabayashi and S. Jegelka, Representation learning on graphs with jumping knowledge networks, in *ICML 2018* (Stockholmsmässan, Stockholm, Sweden, 2018), pp. 5449–5458.
55. C. Chen, K. Petty, A. Skabardonis, P. Varaiya and Z. Jia, Freeway performance measurement system: Mining loop detector data, *Transp. Res. Rec.* **1748** (2001) 96–102.
56. J. Chung, C. Gulcehre, K. Cho and Y. Bengio, Empirical evaluation of gated recurrent neural networks on sequence modeling, in *Advances in Neural Information Processing Systems* (2014), pp. 1–9.
57. S. Hochreiter and J. Schmidhuber, Long short-term memory, *Neural Comput.* **9**(8) (1997) 1735–1780.