Received: 8 September 2020 | Revised: 18 January 2021 | Accepted: 12 February 2021

IET Intelligent Transport Systems

DOI: 10.1049/itr2.12044

The Institution of Engineering and Technology WILEY

ORIGINAL RESEARCH PAPER

Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies

Chenyu Tian | **Wai Kin (Victor) Chan**

Tsinghua-Berkeley Shenzhen Institute, Tsinghua Univeristy, Shenzhen 510006, People's Republic of China

Correspondence
Wai Kin (Victor) Chan, Tsinghua-Berkeley Shenzhen Institute, Tsinghua Univeristy, Shenzhen 510006, Guangdong, People's Republic of China.
Email: chanw@sz.tsinghua.edu.cn

Funding information
the Hylink Digital Solutions Co., Ltd, Grant/Award Number: 120500002; National Natural Science Foundation of China, Grant/Award Number: 71971127

Abstract

Traffic prediction on road networks is highly challenging due to the complexity of traffic systems and is a crucial task in successful intelligent traffic system applications. Existing approaches mostly capture the static spatial dependency relying on the prior knowledge of the graph structure. However, the spatial dependency can be dynamic, and sometimes the physical structure may not reflect the genuine relationship between roads. To better capture the complex spatial-temporal dependencies and forecast traffic conditions on road networks, a multi-step prediction model named Spatial-Temporal Attention Wavenet (STAWnet) is proposed. Temporal convolution is applied to handle long time sequences, and the dynamic spatial dependencies between different nodes can be captured using the self-attention network. Different from existing models, STAWnet does not need prior knowledge of the graph by developing a self-learned node embedding. These components are integrated into an end-to-end framework. The experimental results on three public traffic prediction datasets (METR-LA, PEMS-BAY, and PEMS07) demonstrate effectiveness. In particular, in the 1 h ahead prediction, STAWnet outperforms state-of-the-art methods with no prior knowledge of the network.

1 | INTRODUCTION

With the recent development in intelligent traffic system, the scale and dimension of spatial-temporal data from sensors become larger, which serve as critical inputs to a wide range of applications. Traffic prediction that aims to model the dynamic change of the traffic system is a well-studied spatial-temporal prediction problem, and multi-step traffic forecasting on road network is a crucial task in the transportation industry. High-precision traffic prediction has wide applications. It can not only help travelers plan their routes but also provide insightful information for proactive traffic management strategy to improve traffic efficiency and safety.

The objective of traffic prediction is to predict the future traffic conditions (e.g. traffic volume or speed) in road networks based on historical observations. Many efforts have been conducted to develop methods for traffic prediction [1]. Different from other prediction tasks, traffic prediction on traffic networks need to model the non-Euclidean topology structure of traffic networks, the stochastic characteristic of the time-varying traffic patterns, and spatial-temporal dependencies. This task has two major features. First, there are non-linear temporal correlations, for example, the traffic conditions can fluctuate periodically (e.g. morning peak and evening peak), affecting the correlations between different time steps. Second, there are dynamic spatial correlations, which mean the dependencies of nodes in a road network can change over time considering different traffic conditions, for example, the propagation of the traffic congestion to the upstream and the dissipation.

Recently, deep learning models have been widely applied in traffic prediction and employed in intelligent traffic systems, showing the effectiveness especially when integrating the graph structure into the models [2–8]. Compared with traditional time-series and machine learning methods, deep learning models can flexibly handle relatively long time sequence and large traffic network structure. However, many existing approaches face some major shortcomings.

- Graph convolution and graph attention based methods highly depend on adjacency matrix whose coefficients are

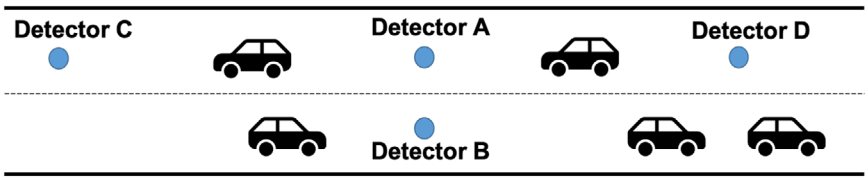


FIGURE 1 A simple example of the distance cannot entail the genuine dependency. Detector A and detector B is close but they are less related because have contrary directions. Detector C and detector D have the same distance to detector A, but they have different effects on it because one is on the upstream and the other is on the downstream

computed by spatial information (e.g. distance between censors), but sometimes the coefficients cannot entail the genuine dependency relationships. To give an example in Figure 1, the close distance between nodes (or detectors) cannot indicate strong spatial dependencies and nodes that have similar distance can have different impacts.

- There are circumstances when connections can not entail the relationship when the connections are missing, for the reason that the spatial information sometimes can be unavailable due to secrecy or privacy, and the relations are more complicated than connectivity considering various attributes such as the number of lanes, surrounding environment, and infrastructure conditions.
- The fixed coefficients may fail to model to dynamic spatial dependencies and result in inaccuracy, because different nodes have different impacts and even the same location has varying influence as time goes by in terms of traffic volume, density and relevant emergent events [2]. They fail to simultaneously model the spatial-temporal features and the dynamic correlations of traffic data.

To address existing challenges, we propose a novel deep learning framework, named as Spatial-Temporal Attention Wavenet (STAWnet). Specifically, we integrate the convolution neural network [9] and attention mechanism [10] into an end-to-end framework to extract the spatial-temporal dependencies. By developing a self-adaptive node embedding, STAWnet can capture the hidden spatial relationship in the data without knowing the graph structure information. We evaluate STAWnet on three public traffic network datasets, METR-LA, PEMS-BAY and PEMS07. STAWnet can achieve satisfactory performance but without prior knowledge of the network as an input, which means our method can be applied to other tasks flexibly. The main contributions of this work are as follows:

- Compared to existing model, STAWnet used self-learned node embedding to learn the latent spatial relationship instead of extracting adjacency relationships from prior knowledge of the graph. It brings high flexibility and can be easily extended to other spatial-temporal forecasting tasks.
- We designed a dynamic attention mechanism that can adjust the coefficients of different nodes based on traffic conditions and spatial information.
- STAWnet can overcome the difficulty in multi-step prediction considering complex dynamic spatial-temporal dependencies and provide certain explainability. The results on real world datasets indicate that STAWnet yields leading predictions performance in terms of various prediction error measures.

The rest of the paper is organized as follows: In Section 2, we give a literature review of related works. In Section 3, we formalize the traffic prediction problem and introduce the overall framework of the STAWnet. In Section 4, experiments are implemented on three datasets to compare with other models. Then we analyze the components of the model in detail in Section 5. Finally, we conclude our work and future directions.

2 | LITERATURE REVIEW

2.1 | Traffic forecasting

Traffic forecasting has been studied for decades, and various emerging methods have been constantly proposed to model traffic characteristics. Lint and Hinsbergen divided these methods into three categories, that is, naive methods, parametric methods and non-parametric methods [11]. Parametric methods often require a wealth of prior knowledge based on queuing theory and traffic flow theory and they cannot handle unpredictability or complex factors. With the rapid development of real-time traffic collection methods, non-parametric (or data-driven) approaches through mass historical data to capture similar traffic patterns prevail in recent years. Further, Zhang et al. divided data-driven methods into three representative subcategories, that is, statistical models, shallow machine learning models and deep learning models [8].

Given historical observations, many traffic prediction studies only consider temporal dependencies using time-series models. The autoregressive integrated moving average (ARIMA) and Kalman filtering have been widely applied [12, 13]. These methods have difficulty achieving high accuracies because they ignore spatial dependencies and only consider the dynamic change of traffic conditions based on the stationary assumption of time sequences. However, this assumption is usually unsatisfied considering traffic dynamics. Machine learning methods such as KNN [14] and SVM [15] are also applied to model complex traffic data and yield satisfactory results. Guo et al. build feature extraction model, and applied k-means method to divide the stations into different types. Then they proposed a hybrid prediction model based on kernel ridge regression and Gaussian process regression to predict the short-term passenger flow of urban rail transit, and verified it on the Automatic Fare Collection System data [16]. However, the performance of traditional machine learning models heavily depends on manual feature

engineering and selection, and they are not suitable for large-scale traffic forecasting.

2.2 | Deep learning on spatial-temporal prediction

In this decade, deep learning methods are prevalent and have achieved high accuracy and efficiency in transportation studies. Ma et al. used the long short-term memory neural network to capture non-linear temporal dynamics effectively [17]. Yang et al. proposed an enhanced long-term features based on LSTM model. It takes full advantages of LSTM in processing time series and overcomes its limitations in insufficient learning of long temporal dependency due to time lag to predict the origin destination flow in the next hour [18], but recurrent neural network models treat traffic sequences of different roads as independent data streams. Following studies further explore the utilities of spatial information. A series of studies applied convolution network to extract spatial patterns by treating inputs as image pixels [19, 20]. Considering spatial-temporal interactions, Sun et al. converted spatial-temporal traffic dynamics to images and applied convolutional neural network (CNN) and recurrent neural network [21]. Chu et al. used multi-scale convolutional long short-term memory network to handle travel demand prediction [22]. Yao et al. further learned the spatial-temporal dependency simultaneously by integrating LSTM, local-CNN and semantic network embedding [23]. Bao et al. developed a hybrid deep learning neural network to predict the short-term demand of free-floating bike sharing [24]. However, the models with deep architectures above do not distinguish spatial variables across topological adjacency. In other words, traffic network has a non-euclidean structure and models based on euclidean structure may compromise the effects of capturing spatial correlations.

Typical deep learning structure like convolutional neural networks cannot be directly used in non-Euclidean and directional structure. To overcome this problem, the graph neural network [25] based techniques have been popular for spatial relationship modeling by aggregating neighboring nodes' information into features. Davis et al studied that graph-based model offers competitive performance against the grid-based model at a lower computational complexity, across three real-world large-scale taxi demand-supply data sets by representing the Voronoi spatial partitions as nodes on an arbitrarily structured graph [26]. As one of the widely used models, graph convolution nework (GCN) [27] methods define graph convolutions by introducing filters from the perspective of graph signal processing, which is based on graph spectral theory and have been applied in related areas. For instance, Yu et al. proposed Spatio-Temporal Graph Convolutional Networks using GCN with Chebyshev polynomial approximation and gated temporal CNN to captures spatial and temporal correlations correspondingly [4]. Li et al. presented the diffusion convolution recurrent neural network [6], which combines diffusion convolution and recurrent neural networks. Wu et al. also adapted diffusion convolution in spatial modelling [5]. It considers both connected and unconnected nodes in the modelling process and uses dilated convolution to learn long sequences of data. Zhang et al. used a multi-graph GCN to capture patterns of passenger inflow and ouflow with different granularity. They combined GCN and a three-dimensional convolutional neural network to predict short-term passenger flow in urban rail transit and achieved leading performance [28]. Jin et al. transferred hybrid GCN model from station-based scenes to grid-based scenes by modelling adjacency matrices and fused graph-level representation and pixel-level representation to obtain joint representation in ride-hailing demand prediction [29]. However, current GCN based models have some shortcomings. They have restrict graph degree and require identical graph structure shared among inputs. Also, they are incapable of learning from topological structure due to fixed graph structure without training. Furthermore, attention based models has been widely applied in deep learning society. It can help neural networks to learn which parts of the input are more relevant. Based on that, graph attention networks employ attention mechanisms which assign larger weights to the more important nodes [10]. It is a promising approach in capturing the correlations between inputs and outputs while improving the interpretability of deep learning models. Do et al. applied spatial and temporal attentions to exploit the spatial dependencies between road segments and temporal dependencies between time steps respectively, which showed promising results and helped to understand spatial-temporal correlations [30]. Guo et al. applied an attention based spatial-temporal graph convolutional networks to effectively capture the dynamic spatial-temporal correlations in traffic data [2]. There are studies took daily and weekly periodic patterns into consideration and used an attention-based periodic-temporal neural network, which captures the spatial, temporal, and periodical correlations [31, 32].

Although these recent works show satisfactory results, graph convolution and graph attention based methods are highly dependent on adjacency matrix whose coefficients are computed by spatial information or context information. Sometimes these coefficients are not given or cannot entail the genuine dependency relationships. Also, the fixed coefficients may fail to model to dynamic dependencies and result in inaccuracy.

3 | METHODOLOGY

In this section, we first give the mathematical definition of the problem. Next, we describe the overall structure and main building blocks of our framework, the temporal convolution layer and the dynamic attention layer. They work together to capture the spatial-temporal dependencies.

3.1 | Problem definition

We define $X(t) \in R^{C*N}$ as the traffic observations at time step t, where C is the number of traffic conditions of interests (e.g. speeds, volumes) and N is the number of detectors in the network. The objective is to learn a function $f(\cdot)$ that map T'

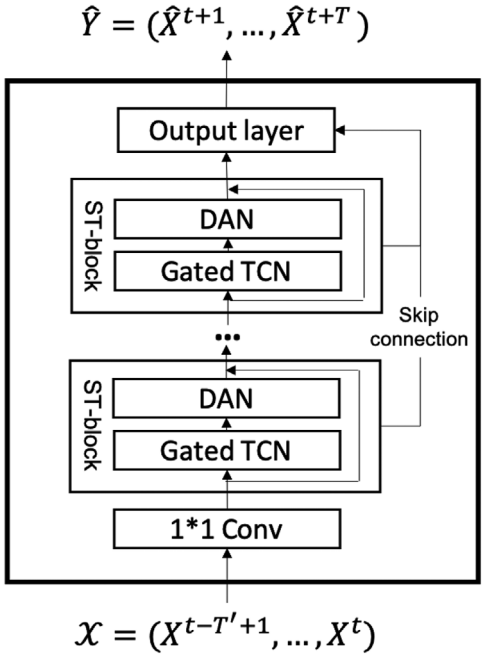


FIGURE 2 The STAWnet consists of multiple ST-blocks. Each ST-block contains a gated TCN and a DAN, where node embedding is integrated into. Layer normalization is utilized within every block to prevent over-fitting. Moreover, both residual and skip connections are used throughout the network to speed up convergence. In the end, the skip outputs from gated TCN in different ST-blocks are added up. Finally, the sum goes through output layers to compute the predictions

historical observations to future T observations as

$$\left[X^{t-T'+1}, \ldots, X^{t}\right] \xrightarrow{f(\cdot)} \left[X^{t+1}, \ldots, X^{t+T}\right]. \tag{1}$$

To clarify, our problem definition is different from most existing relevant studies [2–7, 33]. For others, an input graph must be defined. For example, graph $\mathcal{G}$ with $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$. Here, $\mathcal{V}$ is a set of nodes with $|\mathcal{V}| = N$, $\mathcal{E}$ is a set of edges and $\mathcal{A} \in R^{N*N}$ represents the adjacency matrix. Then the learned function $\tilde{f}(\cdot)$ that map T' historical observations to future T observations as

$$\left[X^{t-T'+1}, \ldots, X^{t}; \mathcal{G}\right] \xrightarrow{\tilde{f}(\cdot)} \left[X^{t+1}, \ldots, X^{t+T}\right]. \tag{2}$$

It is clear to tell that the difference is that our task does not need the graph structure information as an input. This simplification brings high flexibility because sometimes the graph information is unknown or hard to define. Thus, this model needs less prior knowledge and can be easily extend to other spatial-temporal prediction tasks.

3.2 | Framework of the STAWnet

In this section, we elaborate on the proposed architecture of STAWnet. As shown in Figure 2. It consists of multiple stacked spatial-temporal blocks (ST-blocks) and output layers. A ST-block is constructed by a gated temporal convolution network (TCN) and a dynamic attention network (DAN), which are designed to capture the temporal and spatial dependencies correspondingly. By stacking these ST-blocks, it is able to handle spatial-temporal dependencies at different temporal level. The details of each module are described in the following sections.

3.3 | Gated TCN for extracting temporal dependencies

Although RNN-based approaches are prevalent in time-series analysis, they suffer from time-consuming iteration and gradient explosion/vanishing for capturing long-range sequences in practice. CNN-based approaches enjoy the advantages of parallel computing, stable gradients and simple structure. Inspired by [34, 35], we adopt the dilated CNN that allows an exponentially large receptive field by increasing the layer depth aiming to capture temporal dependencies. A dilated convolution is a convolution where the filter is applied over an area larger than its length by skipping input values with a certain step. It is equivalent to a convolution with a larger filter derived from the original filter by dilating it with zeros, but is more efficient [35]. As a special case, dilated convolution with dilation 1 yields the standard convolution. In addition, a gated mechanism is used to learn complex temporal dependencies. The computation in Gated TCN is written as

$$\mathcal{X}_T^l = \tanh\left(W_{f,l} * \mathcal{X}_{out}^{l-1}\right) \odot \sigma\left(W_{g,l} * \mathcal{X}_{out}^{l-1}\right), \tag{3}$$

where $\mathcal{X}_{out}^{i} \in \mathbf{R}^{C^i \times N \times T^i}$ is the output of the i^{th} ST-block, C^i is the number of channels of the input data in the i^{th} ST-block, and T^i is the length of the temporal dimension in the i^{th} ST-block. σ is the sigmoid activation function, $\odot$ denotes an element-wise multiplication operator, $*$ denotes a convolution operator, f and g denote filter and gate, and W is the learnable convolution filter.

Given inputs from last layer as three-dimension tensors with size $[C^{l-1}, N, T^{l-1}]$. After the gate TCN, the outputs become three-dimension tensors with size $[C^l, N, T^l]$ with $T^l = T^{l-1} - d^l$, where d^l is the dilation size in the l^{th} ST-block. Thus, the length of the temporal dimension of the tensors get shorter after going through gated TCN layers. For the input of the first ST-block, $\mathcal{X}_{out}^{0} = \mathrm{Conv}_{1*1}(\mathcal{X}) \in \mathbf{R}^{C^0 \times N \times T'}$, where $Conv_{1*1}$ is the 1*1 convolution computation which is used to increase dimensionality.

3.4 | Attention mechanism for extracting dynamic spatial dependencies

The impacts from other roads can be dynamic considering different traffic conditions. For example, the impact of one road to other roads can be stronger when the road is congested compared to when it has low volume [36]. To better model the dynamic spatial dependencies, the self-attention network is employed on graph-structured data to extract patterns in

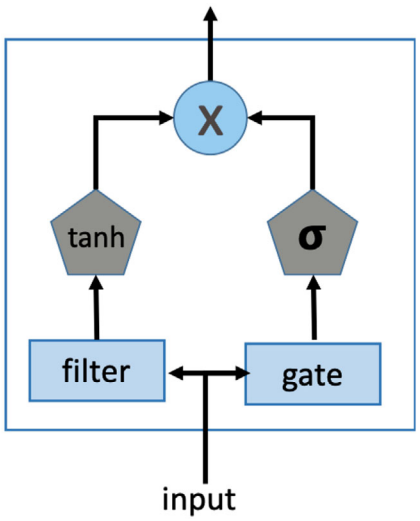


FIGURE 3 Framework of Gated TCN

our model accordingly. Attention mechanism has been widely applied in deep learning society due to their high efficiency and flexibility in modelling dependencies and achieved good results in different tasks like computer vision [37], natural language processing [38], and graph learning [10].

The key idea of attention is to dynamically assign different weights to different nodes, as shown in Figure 4. For node i, we compute a weighted sum form all other nodes' information in the network:

$$h'^{l}_{i,t} = \sum_{j \le N} \alpha_{i,j} \cdot h^{l}_{j,t}, \tag{4}$$

where $\alpha_{i,j}$ is the attention score indicating the importance of node j to node i with $\sum_{j \le N} \alpha_{i,j} = 1$, $h^{l}_{i,t} = \mathcal{X}^{l}_{T}[:, i, t]$ and $h^{l}_{i,t} \in \mathbf{R}^{C^l}$.

From real-world experience and traffic flow studies [39], both the traffic network structure and traffic conditions could help to predict future conditions. Motivated by this intuition, we incorporate both the intrinsic network information and the traffic condition into the prediction models. So the self-learned node embedding is concatenated with the hidden state, and adopt the scaled dot-product approach to compute the attention. Considering the aforementioned complex factors affecting the relationships between nodes, we propose to learn a node embedding to capture the hidden representations of every node in the network. Node embedding is a mapping of a discrete node ID to a vector of continuous numbers. In other words, embeddings are low-dimensional, learned continuous vector representations of discrete variables (node ID). In other words, it projects the nodes into vectors with latent information like famous Word2Vec model [40]. In practice, it is randomly initialized and gradually trained. The well-trained embeddedings are representations of nodes where similar nodes are closer to one another. Neural network embeddings are useful because they can reduce the dimensionality of categorical variables and meaningfully represent categories in the transformed space. Following, we compute

$$o_{i,j} = \frac{\langle W_q(h^{l}_{i,t} \| e_i), W_k(h^{l}_{j,t} \| e_j) \rangle}{\sqrt{d'}}, \tag{5}$$

where $\|$ represents the concatenation operation, $\langle \cdot, \cdot \rangle$ denotes the inner product operation, e_i is the node embedding of the

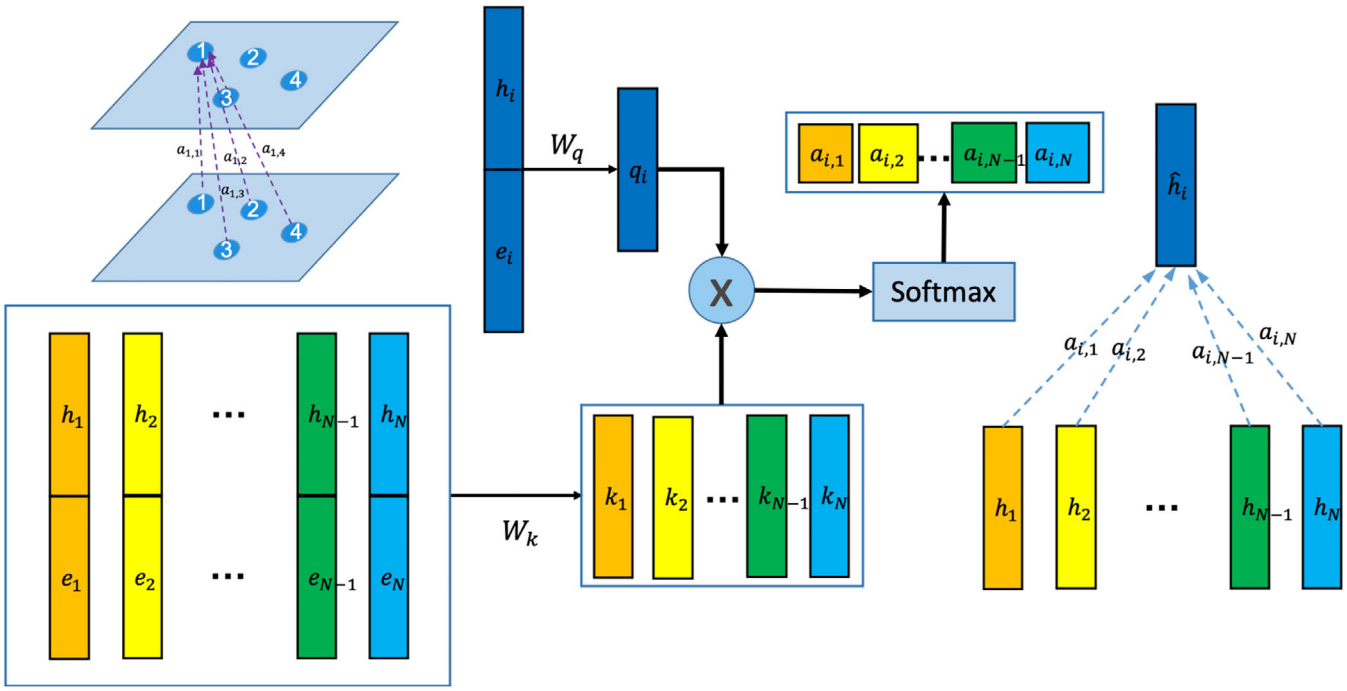


FIGURE 4 Framework of DAN

node i, $W_k, W_q \in \mathbb{R}^{d' \times d_c}$ are the key and query matrix in DAN, and d_c is the dimension of $h^l_{i,t} \| e_i$. The node embedding e_i, the key matrix W_k, and query matrix W_q are all learnable parameters.

$$\alpha_{i,j} = \frac{\exp(o_{i,j})}{\sum_{k \le N} \exp\left(o_{i,k}\right)}. \tag{6}$$

After the attention scores are obtained, the hidden state can be updated through Equation (4). The shape of the outputs is the same as inputs with $\mathcal{X}^l_S[:, i, t] = h'^l_{i,t}$. This operation is efficient because it is parallelizable across node pairs.

3.5 | The output layer

ResNet mechanism is applied in the framework to prevent gradient vanishing [41]. The output of the l^{th} ST-block is computed as

$$\mathcal{X}^l_{out} = \mathcal{X}^l_S + \mathcal{X}^{l-1}_{out}. \tag{7}$$

In every ST-block, the gated TCN has a skip output as shown in 2. Then the skip connections are added up as $\sum_{i \le N_{st}} Conv_{1*1}(\mathcal{X}^i_T)$, where N_{st} is the number of ST-blocks. Following, two layers of non-linear transformation with ReLU [42] as the activation function are used to compute the final output. During the training, the goal is to minimize the error between real traffic observations on the roads and the predicted value. The loss functions is shown in Equation (8).

$$\text{Loss} = \sum_{\tau=t+1}^{t+T} \left\| X^\tau - \hat{X}^\tau \right\|, \tag{8}$$

with

$$\left[\hat{X}^{t+1}, \ldots, \hat{X}^{t+T}\right] = f(X^{t-T'+1}, \ldots, X^t), \tag{9}$$

and $f(\cdot)$ is the prediction model.

4 | EXPERIMENTS

4.1 | Datasets

We evaluate the performance of our model and baseline models on three widely-used traffic prediction datasets with different road network scales:

1. Traffic speed prediction on the METR-LA dataset, which contains 4 months of data recorded by 207 loop detectors ranging from 1 March 2012 to 30 June 2012 in the highway of Los Angeles.
2. Traffic speed prediction on the PEMS-BAY dataset, which contains 6 months of data recorded by 325 sensors ranging from 1 January 2017 to 30 June 2017 in the Bay Area.
3. Traffic flow prediction on the PEMS07 dataset, which contains 4 months of data recorded by 883 sensors ranging from 1 May 2017 to 31 August 2017.

4.2 | Benchmarks

To demonstrate the effectiveness of the proposed model, We compare STAWnet with the following models:

- HA: Historical average, which is a naive method that models the traffic flow as a periodic process and uses the weighted average of previous periods as the prediction.
- ARIMA: Auto-Regressive Integrated Moving Average Model, which is a classical time series prediction model.
- FC-LSTM: Recurrent neural network with fully connected LSTM hidden units [43].
- T-GCN: Temporal GCN [7] combines the graph convolution network and gated recurrent unit.
- DCRNN: Diffusion Convolutional Recurrent Neural Network [6], which combines recurrent neural networks with diffusion convolution modeling both inflow and outflow relationships.
- STGCN: Spatial-Temporal Graph Convolution Network [4], which applies purely convolutional structures to extract spatial-temporal features simultaneously from graph-structured time series.
- GaAN: Gated Attention Networks [44], uses a multi-head attention-based network with a convolutional sub-network to control each attention head's importance.
- Graph WaveNet: A convolution network architecture [5], which introduces a self-adaptive graph to capture the hidden spatial dependency, and uses dilated convolution to capture the temporal dependency.
- APTN: Attention-based Periodic-Temporal neural Network [31], which is an end-to-end solution for traffic forecasting that captures spatial, short-term, and long-term periodical dependencies.
- ST-GRAT: Spatiao-Temporal GRaph ATtention [33], which uses spatial attention, temporal attention, and spatial sentinel vectors to capture the spatiotemporal dynamic in road networks.

For all mentioned approaches, the input are uniformed and the hyperparameters are tuned that performed best on the validation set.

4.3 | Experimental settings

Following the previous works [4–6], We use $T = T' = 12$ with historical and prediction timesteps (1 h). The dataset is split into three parts with 70% of the data used for training, 20% used

for test and 10% used for validation. We train the model using Adam optimizer [45] with learning rate of 0.001, batch size of 64, and epochs of 100. The training objective is L1 loss. The dimension of each node embedding vector is 16, hidden dimension is 32. The number of ST-blocks is 8. Our experimental platform is on the server with eight CPUs (Intel(R) Xeon(R) Gold 6254 CPU @ 3.10GHz), 256-GB RAM, and two GPUs (NVIDIA GeForce RTX 2080 Ti, 11GB memory).

4.4 | Experimental results

In the experiment, we measure the accuracy of the models using mean absolute error (MAE), root mean squared error (RMSE), and mean absolute percentage error (MAPE) computed as

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y-\hat{y}}{y}\right|$$
$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y-\hat{y}| \quad . \tag{10}$$
$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y-\hat{y})^2}$$

Table 1 shows the metrics of the STAWnet and the baseline algorithms for 5, 15, 30 and 60 min ahead forecasting on the test datasets. Time series analysis method is not ideal, sometimes even worse than historical average method in multi-steps, indicating its limited abilities of modeling non-linear and complex spatial-temporal data, but LSTM as a deep learning method obtains better prediction results compared with ARIMA. The models which consider both the spatial and temporal relationships, including DCRNN, STGCN, Graph WaveNet and ST-GRAT, achieve satisfactory results. Although the accuracy of the STAWnet is slightly lower than Graph WaveNet and ST-GRAT in the shorter-than-30-min prediction on the first two dataset, it drops much slower as the prediction sequence getting longer and STAWnet is more accurate in the long-time prediction. Considering the average metrics on all the horizons, the STAWnet achieves the most accurate performance as shown in the Table 2.

For the multi-step prediction task, it is worth noticing that the accuracy of the long-time prediction is of high importance no matter in theory or in practice. Because the long-time prediction usually is the bottleneck of the model accuracy with the existence of error propagation and historical knowledge forgetting. A more accurate long-time prediction can give reliable guidance for travel planning, departure time scheduling, and traffic regulation.

Because the performance gap on these two datasets is minor, a more challenging dataset PEMS07 with 883 detectors is added. On PEMS07, STAWnet outperforms the other benchmarks in terms of almost all metrics. On 60-minute prediction, STAWnet achieves approximately 5.3% higher performance in terms of MAE, 2.8% higher in terms of RMSE and 7.0% higher in terms of MAPE, showing effectiveness on capture complex spatiotemporal dependencies. It is worth noticing that STAWnet still achieves better results without being given the information of the detector network, such as distance, functional similarity, and connectivity. The performance will drop dramatically (e.g. Graph WaveNet, ST-GRAT) or the model does not even work (e.g. DCRNN, STGCN) for other models without this information as input. Thus, the intuition behind the experiments is that STAWnet is more flexible and accurate compared with existing models. STAWnet is slightly worse on the short-time prediction because the information from adjacent nodes is more useful in short-time prediction, but STAWnet does not use underlying spatial structures. Actually, STAWnet has more strength in the long-time prediction and is more flexible compared with baselines, and on average, the overall accuracy of STAWnet is the best.

4.5 | Computation speed

Select the leading performance models. In this section, we report the computation costs of the models on the METR-LA dataset, as shown in Table 3. Comparing STAWnet to the baselines, we find it is about four times faster than GaAn. STAWnet is also faster than DCRNN, ST-GRAT and APTN. The Graph WaveNet performs best because it is a non-autoregressive model. Overall, STAWnet is the second best model in terms of the training time cost and inference time cost. On the other hand, Graph WaveNet, DCRNN and ST-GRAT need graph information and APTN needs periodic observations as input. Correspondingly, extra data preprocessing is also needed, but STAWnet does not need auxiliary information. Overall, STAWnet is faster and more flexible to similar tasks without too much data preprocessing.

5 | MODEL ANALYSIS

5.1 | Self-learned node embedding

We further investigate the relationship of the nodes on the METR-LA experiment. Shown in Figure 5(a) as a heatmap, the real adjacency relationship is based on the distance between nodes, and the heatmap value W_{ij}^{d} is computed as

$$W_{ij}^{d} = \exp\left(-\frac{\text{dist}\left(v_i, v_j\right)^2}{\sigma^2}\right) \text{if dist}\left(v_i, v_j\right) \leq \kappa_d, \text{ otherwise } 0, \tag{11}$$

where W_{ij}^{d} represents the edge weight between sensor v_i and v_j calculated by dist(v_i, v_j), which denotes the euclidean distance between sensor v_i and v_j, σ is the standard deviation of the distances and κ_d is the distance threshold [6]. As shown in the Figure 5(b), the heatmap value of the self-learned adjacency W_{ij}^{s} is based on the cosine similarity between node embedding

TABLE 1 Summary of experiment results on the datasets in terms of multiple metrics (some results are omitted because the codes are not open-sourced)

		5 min			15 min			30 min			60 min		
Data	**Models**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
METR-LA	HA	4.16	7.80	13.0%	4.16	7.80	13.0%	4.16	7.80	13.0%	4.16	7.80	13.0%
	ARIMA	3.77	7.66	11.2%	3.99	8.21	9.60%	5.15	10.45	12.70%	6.90	13.23	17.40%
	FC-LSTM	3.01	5.92	8.24%	3.44	6.30	9.60%	3.77	7.23	10.90%	4.37	8.69	13.20%
	T-GCN	2.39	4.23	6.02%	3.03	5.26	7.81%	3.52	6.12	9.45%	4.30	7.31	11.8%
	DCRNN	**2.18**	**3.77**	**5.17%**	2.77	5.38	7.30%	3.15	6.45	8.80%	3.60	7.60	10.50%
	STGCN	2.31	4.04	5.45%	2.88	5.74	7.62%	3.47	7.24	9.57%	4.59	9.40	12.70%
	GaAN	-	-	-	2.71	5.24	6.99%	3.12	6.36	8.56%	3.64	7.65	10.62%
	Graph WaveNet	2.22	3.83	5.33%	2.69	5.15	6.90%	3.07	6.22	8.37%	3.53	7.37	10.01%
	APTN	2.30	4.02	5.67%	2.76	5.38	7.30%	3.15	6.43	8.80%	3.70	7.69	10.69%
	ST-GRAT	-	-	-	**2.60**	**5.07**	**6.61%**	**3.01**	**6.21**	**8.15%**	3.49	7.42	10.01%
	STAWnet	2.28	3.99	5.61%	2.70	5.22	6.98%	3.04	6.14	8.22%	**3.44**	**7.16**	**9.82%**
PEMS-BAY	HA	2.88	5.59	6.8%	2.88	5.59	6.8%	2.88	5.59	6.8%	2.88	5.59	6.8%
	ARIMA	1.45	3.19	3.14%	1.62	3.30	3.50%	2.33	4.76	5.40%	3.38	6.50	8.30%
	FC-LSTM	1.85	3.50	4.16%	2.05	4.19	4.80%	2.20	4.55	5.20%	2.37	4.96	5.70%
	T-GCN	1.12	1.60	2.29%	1.50	2.83	3.14%	1.73	3.40	3.76%	2.18	4.35	4.94%
	DCRNN	**0.85**	**1.54**	**1.63%**	1.38	2.95	2.90%	1.74	3.97	3.90%	2.07	4.74	4.90%
	STGCN	0.86	1.56	1.66%	1.36	2.96	2.90%	1.81	4.27	4.17%	2.49	5.69	5.79%
	GaAN	-	-	-	-	-	-	-	-	-	-	-	-
	Graph WaveNet	**0.85**	**1.54**	**1.63%**	1.30	2.74	2.73%	1.63	3.70	3.67%	1.95	4.52	4.63%
	APTN	0.99	1.86	1.77%	1.38	2.96	2.91%	1.97	3.95	3.69%	2.33	4.60	4.65%
	ST-GRAT	-	-	-	**1.29**	**2.71**	**2.67%**	**1.61**	**3.69**	**3.63%**	1.95	4.54	4.64%
	STAWnet	0.86	1.56	1.68%	1.31	2.78	2.76%	1.62	3.70	3.67%	**1.89**	**4.36**	**4.47%**
PEMS07	HA	28.48	52.58	12.0%	28.48	52.58	12.0%	28.48	52.58	12.0%	28.48	52.58	12.0%
	ARIMA	21.19	34.22	8.98%	23.44	37.23	10.43%	32.29	50.22	14.26%	40.97	60.07	16.75%
	FC-LSTM	21.86	34.94	9.29%	24.66	39.25	10.52%	29.98	45.84	13.20%	35.12	54.07	15.65%
	T-GCN	18.19	28.92	8.18%	21.86	34.94	9.29%	25.38	38.78	11.08%	30.12	48.33	13.89%
	DCRNN	17.21	27.07	7.32%	21.03	32.83	8.92%	23.94	37.05	10.28%	29.49	44.59	13.07%
	STGCN	17.47	27.26	7.80%	22.59	33.81	8.96%	24.00	37.14	10.30%	29.63	44.78	13.11%
	GaAN	-	-	-	-	-	-	-	-	-	-	-	-
	Graph WaveNet	**16.75**	**26.79**	**7.21%**	19.06	30.89	8.05%	20.74	33.53	8.84%	23.44	37.23	10.43%
	APTN	17.92	28.10	7.75%	20.13	31.88	8.21%	21.91	34.78	8.78%	24.30	38.58	10.47%
	ST-GRAT	-	-	-	-	-	-	-	-	-	-	-	-
	STAWnet	16.97	27.13	7.69%	**18.74**	**30.56**	**7.94%**	**20.09**	**32.99**	**8.45%**	**22.21**	**36.18**	**9.70%**

computed as

$$W_{ij}^{s} = \frac{e_i \cdot e_j}{\|e_i\|\|e_j\|} \quad \text{if } W_{ij} \geq \kappa_s, \text{ otherwise } 0, \tag{12}$$

where κ_s is the similarity threshold. It needs to be mentioned that we choose two different metrics to evaluate these two adjacencies because v_i, which is a coordinate, and e_i, which is a node embedding, are different types of data having different dimensions and properties. It is more suitable to choose appropriate metrics for them.

As qualitative descriptions, although no information of the sensor positions is given, the relationship heatmap of the self-learned node embedding obtains some similar patterns as the real-world adjacency heatmap, which are labelled in red rectangles. On the other hand, the difference of these two heatmaps is that the self-learned relationship have more bluer points in the figure, indicating that the self-learned adjacency heatmap can give more hidden relationships with other nodes, not just limited to the close neighbors. The effectiveness of self-learned node embedding improving the prediction accuracy are further quantitatively discussed in the following.

TABLE 2 The overall performance on the METR-LA and PEMS-BAY

Dataset	Model	MAE	RMSE	MAPE
METR-LA	Graph WaveNet	3.09	6.26	8.42%
	ST-GRAT	3.03	6.23	8.25%
	STAWnet	**3.01**	**6.03**	**8.14%**
PEMS-BAY	Graph WaveNet	1.63	3.65	3.67%
	ST-GRAT	1.62	3.65	3.65%
	STAWnet	**1.56**	**3.48**	**3.51%**

TABLE 3 The computation times on the METR-LA dataset

Computation time	DCRNN	GaAN	Graph Wavenet	APTN	ST-GRAT	STAW net
Training (s/epoch)	504.4	1461.4	203.9	843.4	341.7	302.1
Inference (s)	34.0	131.1	8.4	143.2	48.7	13.3

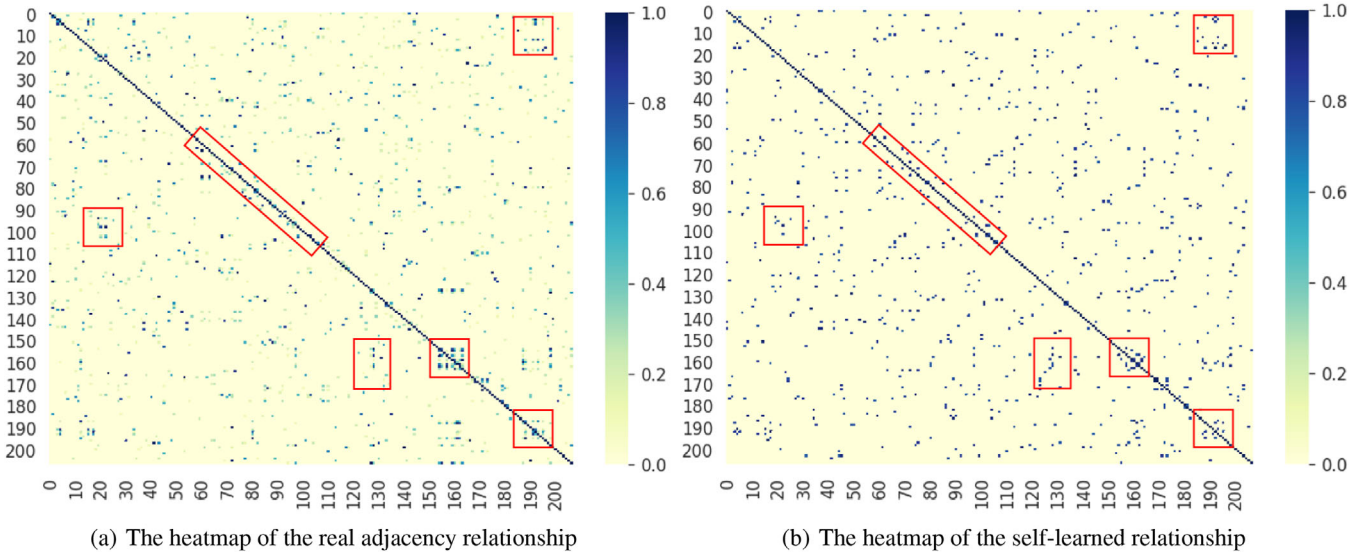
TABLE 4 Experiment results of different attention configurations

Dataset	Model	MAE	RMSE	MAPE
METR-LA	DAN w/o	3.58	7.18	10.21%
	Dynamic w/o	3.05	6.09	8.25%
	Node embedding w/o	3.52	7.03	10.18%
	STAWnet	**3.01**	**6.03**	**8.14%**
PEMS-BAY	DAN w/o	1.80	4.06	4.16%
	Dynamic w/o	1.56	3.49	3.52%
	Node embedding w/o	1.76	3.97	4.09%
	STAWnet	**1.56**	**3.48**	**3.51%**
PEMS07	DAN w/o	26.94	42.24	12.86%
	Dynamic w/o	20.11	32.87	8.64%
	Node embedding w/o	22.89	36.99	9.56%
	STAWnet	**20.07**	**32.78**	**8.55%**

5.2 | Ablation studies

To quantitatively verify the effectiveness of the model design, we conduct experiments with different configurations. The "DAN w/o" means the model do not include the dynamic attention network. The "dynamic w/o" means the model only use the node embedding to do the attention computation without concatenating the output from the TCN, which can be seen as a fixed spatial dependency. The "node embedding w/o" case means the model only use output from the TCN to do attention without node embedding, which can be seen as only considering the similarity of dynamic historical observations between nodes without the learned knowledge of the network structure. Table 4 shows the average score of MAE, RMSE and MAPE over 1-h prediction.

Compared with STAWnet, the "DAN w/o" experiment not only shows the effectiveness of DAN in capturing the spatial dependencies and improving prediction accuracy, but also gives the information that gated TCN is more effective in capturing temporal relationship compared with LSTM (the MAEs of "DAN w/o" on 5, 15, 30, and 60 min prediction are 2.28, 2.99, 3.59, and 4.45 on METR-LA dataset, 0.86, 1.40, 1.84, and 2.36 on PEMS-BAY dataset, and are 18.20, 21.99, 26.21, and 35.41 on PEMS07 dataset). When implementing the attention, the node embedding and the output from Gated TCN are concatenated. From the results, it can be concluded that the self-leaned node embedding greatly helps to improve the accuracy, and dynamic attention mechanism helps to achieve superior results than baselines. The outputs from gated TCN concatenating with self-learned node embedding are able to assign dynamic weights to different neighbors and better depict the



(a) The heatmap of the real adjacency relationship (b) The heatmap of the self-learned relationship

FIGURE 5 The comparison between the real adjacency relationship and the self-learned relationship on the METR-LA dataset. The self-learned adjacency matrix has similar patterns as the real adjacency matrix

(a) The attention distribution of the node 769373 on METR-LA map.

(b) The details of the attention over nearby nodes.

FIGURE 6 Test

spatial-temporal dependencies by considering both the network structure and the traffic conditions. Comparing GCN-based models with adjacency matrix, which can be seen as a static relationship, and sometimes the static relationship can be misleading. For example, for relatively long-time prediction, the observations from distant nodes can be more useful compared with adjacent nodes. Without the constraints of the graph, DAN with spatial attention can selectively focus on certain nodes rather than treating all adjacent nodes equally, and this data-driven attention mechanism captures the dynamic correlations as well. These nodes with different weights can reconstruct the useful relationship of the entire graph. Also, considering all the nodes brings regularization effect to the model and avoids over-fitting between nodes.

FIGURE 7 The dynamic attention distribution of the node 717583 on METR-LA map

5.3 | Attention interpretation and visualization

The attention mechanism can also give some explanations about how the model learns the spatial dependencies from others. To better understand the contribution of the attention weights, we visualize representative attention weights in Figure 6(a) and their physical locations on the real map to show how the proposed model can handle complex traffic situations.

To illustrate, we randomly select a traffic sensor node (sensor id: 769373) in the METR-LA dataset. The darkness of the colors represents the magnitude of attention. Based on Figure 6(a), we can find most nodes with most dark colors located at the regions which are close to the sensor 769373. It is reasonable because the traffic condition of close neighbors have most influence to predict future traffic speed. Another area with higher attention is located between the downstream nodes and the freeway intersection in the south which leads to the downtown of the Los Angeles. Also, the model also gives relatively high attention to the north-west corner and south-east corner, which are both traffic-busy areas in the city. Zooming in to find details in Figure 6(b), it is interesting to tell that the nodes on the same side with same vehicular directions have more importance compared with nodes with a contrast vehicular direction when making predictions, which strongly supports the aforementioned description in Figure 1.

The attention of nodes is also dynamic in terms of different traffic conditions. Take sensor node (sensor id: 717583) as another example. As shown in the Figure 7, at first, the model gives more attention to nodes in the downstream at morning peak hour. After 30 min, the attentions are transferred to both upstream and downstream nodes. After 1 h, the model more focuses on the upstream node, and some of the nodes near Node 717583 with light color are in the contrary direction, so they are given less importance.

It can concluded that the self-learned node embedding gives the latent information of nodes. In the prediction process, the attention part captures the dynamic spatial-temporal dependencies. Hence, STAWnet not only achieves a state-of-the-art forecasting performance but also shows an interpretability advantage to a certain extent.

6 | CONCLUSIONS

To summarize, the dynamic change of traffic conditions on road network exhibits spatial and temporal dependencies. This paper presents STAWnet to capture spatial-temporal dependencies efficiently by combining temporal convolution with attention mechanism. The design of the self-learned node embedding gives an insight that prior knowledge of the graph may not be necessary in learning the spatial-temporal dependencies for traffic prediction tasks, and the dynamic dependencies can help to improve the prediction accuracy. Experiments on three real-world datasets show that STAWnet achieves state-of-the-art results with fewer inputs compared with related studies. In addition, the self-learned node embedding and attention weights can help to identify the influential nodes indicating the interpretability of the proposed model. Our source code are available at https://github.com/CYBruce/STAWnet.

For future works, a more detailed analysis of spatial-temporal patterns at road networks should be analyzed to find the explainable spatial-temporal patterns, and we will further explore the utilities of external context data, such as venue types, weather conditions and event information as multi-view graph to other spatial-temporal prediction tasks. Also, the self-learned node embedding can be transferred to other related tasks like data imputing, clustering analysis and relationship identification.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China (Grant No. 71971127) and the Hylink Digital Solutions Co., Ltd. (120500002).

ORCID

Chenyu Tian https://orcid.org/0000-0002-7748-5873
Wai Kin (Victor) Chan https://orcid.org/0000-0002-7202-1922

REFERENCES

1. Ermagun, A., Levinson, D.: Spatiotemporal traffic forecasting: review and proposed directions. Transp. Rev. 38(6), 786–814 (2018)
2. Song, C., et al.: Attention based spatial-temporal graph convolutional networks for traffic flow forecasting. In: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI-20, pp. 662–669 (2020)
3. Zheng, C., et al.: Gman: A graph multi-attention network for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI-20, pp. 1234–1241 (2020)
4. 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. (International Joint Conferences on Artificial Intelligence Organization), pp. 3634–3640 (2018)
5. Wu, Z., et al.: Graph wavenet for deep spatial-temporal graph modeling. In: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19. (International Joint Conferences on Artificial Intelligence Organization), pp. 1907–1913 (2019)
6. Li, Y., et al.: Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In: International Conference on Learning Representations (ICLR '18) (2018)
7. Zhao, L., et al.: T-gcn: A temporal graph convolutional network for traffic prediction. In: IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 9, pp. 3848–3858 (2019)
8. Zhang, Z., et al.: Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. Transp. Res. C, Emerg. Technol. 105, 297–322 (2019)
9. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature 521(7553), 436–444 (2015)
10. Veličković, P., et al.: Graph Attention Networks, International Conference on Learning Representations, https://openreview.net/forum?id=rJXMpikCZ (2018)
11. Van-Lint, J., Van-Hinsbergen, C.: Short-term traffic and travel time prediction models. In: Transportation Research Circular E-C168: Artificial Intelligence Applications to Critical Transportation Issues, pp. 22–41. Transportation Research Board (2012)
12. Lippi, M., Bertini, M., Frasconi, P.: Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. IEEE Trans. Intell. Transp. Syst. 14(2), 871–882 (2013)
13. Moreira-Matias, L., et al.: Predicting taxi–passenger demand using streaming data. IEEE Trans. Intell. Transp. Syst. 14(3), 1393–1402 (2013)
14. Cai, P., et al.: A spatiotemporal correlative k-nearest neighbor model for short-term traffic multistep forecasting. Transp. Res. C, Emerg. Technol. 62, 21–34 (2016)
15. Tang, J., et al.: Traffic flow prediction based on combination of support vector machine and data denoising schemes. Phys. A 534, 120642 (2019)
16. Guo, Z., et al.: Short-term passenger flow forecast of urban rail transit based on gpr and krr. IET Intell. Transp. Syst. 13(9), 1374–1382 (2019)
17. Ma, X., et al.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. C, Emerg. Technol. 54, 187–197 (2015)
18. Yang, D., et al.: Urban rail transit passenger flow forecast based on lstm with enhanced long-term features. IET Intell. Transp. Syst. 13(10), 1475–1482 (2019)
19. Li, J., et al.: Deep neural network for structural prediction and lane detection in traffic scene. IEEE Trans. Neural Netw. Learn. Syst. 28(3), 690–703 (2016)
20. Ma, X., et al.: Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. Sensors 17(4), 818 (2017)
21. Sun, S., Chen, J., Sun, J.: Traffic congestion prediction based on gps trajectory data. Int. J. Distrib. Sensor Netw. 15(5), 1550147719847440 (2019)
22. Chu, K.F., Lam, A.Y., Li, V.O.: Deep multi-scale convolutional lstm network for travel demand and origin-destination predictions. IEEE Trans. Intell. Transp. Syst. 21(8), 3219–3232 (2019)
23. Yao, H., et al.: Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence, AAAI-19, pp. 5668–5675 (2019)
24. Bao, J., Yu, H., Wu, J.: Short-term ffbs demand prediction with multi-source data in a hybrid deep learning framework. IET Intell. Transp. Syst. 13(9), 1340–1347 (2019)
25. Zhou, J., et al.: Graph neural networks: A review of methods and applications. arXiv preprint arXiv:181208434 (2018)
26. Davis, N., Raina, G., Jagannathan, K.: Grids versus graphs: Partitioning space for improved taxi demand-supply forecasts. In: IEEE Transactions on Intelligent Transportation Systems (2020)
27. Bruna, J., et al.: Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:13126203 (2013)
28. Zhang, J., et al.: Multi-graph convolutional network for short-term passenger flow forecasting in urban rail transit. IET Intell. Transp. Syst. 14(10), 1210–1217 (2020)
29. Jin, G., et al.: Urban ride-hailing demand prediction with multiple spatio-temporal information fusion network. Transp. Res. C, Emerg. Technol. 117, 102665 (2020)
30. Do, L.N., et al.: An effective spatial-temporal attention based neural network for traffic flow prediction. Transp. Res. C, Emerg. Technol. 108, 12–28 (2019)
31. Shi, X., et al.: A spatial-temporal attention approach for traffic prediction. In: IEEE Transactions on Intelligent Transportation Systems (2020)
32. Wu, Y., et al.: A hybrid deep learning based traffic flow prediction method and its understanding. Transp. Res. C, Emerg. Technol. 90, 166–180 (2018)
33. Park, C., et al.: St-grat: A novel spatio-temporal graph attention networks for accurately forecasting dynamically changing road speed. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. CIKM '20. New York, NY, USA: Association for Computing Machinery, 1215–1224 (2020). https://doi.org/10.1145/3340531.3411940
34. Van den Oord, A., et al.: Conditional image generation with pixelcnn decoders. In: Advances in Neural Information Processing Systems, 4790–4798 (2016)
35. van den Oord, A., et al.: Wavenet: A generative model for raw audio. CoRR, http://arxiv.org/abs/1609.03499 (2016)
36. Salamanis, A., et al.: Managing spatial graph dependencies in large volumes of traffic data for travel-time prediction. IEEE Trans. Intell. Transp. Syst. 17(6), 1678–1687 (2016)
37. Li, L., et al.: Attention based glaucoma detection: A large-scale database and CNN model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10571–10580 (2019)
38. Vaswani, A., et al.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)
39. Pan, T., et al.: Short-term traffic state prediction based on temporal–spatial correlation. IEEE Trans. Intell. Transp. Syst. 14(3), 1242–1254 (2013)
40. Mikolov, T., et al.: Efficient estimation of word representations in vector space, http://arxiv.org/abs/1301.3781 (2013)
41. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition, Las Vegas, NV, 2016, pp. 770–778

42. Li, Y., Yuan, Y.: Convergence analysis of two-layer neural networks with relu activation. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., et al., editors. Advances in Neural Information Processing Systems 30. Curran Associates, Inc., pp. 597–607 (2017)
43. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Ghahramani, Z., et al. (eds.) Advances in Neural Information Processing Systems 27. Curran Associates, New York, pp. 3104–3112 (2014)
44. Zhang, J., et al.: Gaan: Gated attention networks for learning on large and spatiotemporal graphs. arXiv preprint arXiv:180307294 (2018)
45. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization, CoRR, abs/1412.6980 (2015)

How to cite this article: Tian C, Chan WK(V). Spatial-temporal attention wavenet: A deep learning framework for traffic prediction considering spatial-temporal dependencies. *IET Intell Transp Syst.* 2021;15:549–561. https://doi.org/10.1049/itr2.12044