



TARGCN: temporal attention recurrent graph convolutional neural network for traffic prediction

He Yang¹ · Cong Jiang² · Yun Song² · Wendong Fan³ · Zelin Deng² · Xinke Bai²

Received: 5 April 2024 / Accepted: 24 July 2024
© The Author(s) 2024

Abstract

Traffic prediction is crucial to the intelligent transportation system. However, accurate traffic prediction still faces challenges. It is difficult to extract dynamic spatial–temporal correlations of traffic flow and capture the specific traffic pattern for each sub-region. In this paper, a temporal attention recurrent graph convolutional neural network (TARGCN) is proposed to address these issues. The proposed TARGCN model fuses a node-embedded graph convolutional (Emb-GCN) layer, a gated recurrent unit (GRU) layer, and a temporal attention (TA) layer into a framework to exploit both dynamic spatial correlations between traffic nodes and temporal dependencies between time slices. In the Emb-GCN layer, node embedding matrix and node parameter learning techniques are employed to extract spatial correlations between traffic nodes at a fine-grained level and learn the specific traffic pattern for each node. Following this, a series of gated recurrent units are stacked as a GRU layer to capture spatial and temporal features from the traffic flow of adjacent nodes in the past few time slices simultaneously. Furthermore, an attention layer is applied in the temporal dimension to extend the receptive field of GRU. The combination of the Emb-GCN, GRU, and the TA layer facilitates the proposed framework exploiting not only the spatial–temporal dependencies but also the degree of interconnectedness between traffic nodes, which benefits the prediction a lot. Experiments on public traffic datasets PEMS03, PEMS04, and PEMS08 demonstrate the effectiveness of the proposed method. Compared with state-of-the-art baselines, it achieves 4.62% and 5.78% on PEMS03, 3.08% and 0.37% on PEMS04, and 5.08% and 0.28% on PEMS08 superiority on average. Especially for long-term prediction, prediction results for the 60-min interval show the proposed method presents a more notable advantage over compared benchmarks. The implementation on Pytorch is publicly available at <https://github.com/csust-sonie/TARGCN>.

Keywords Intelligent transportation · Traffic prediction · Deep learning · Graph convolutional neural network

Introduction

Many countries have made great efforts to develop intelligent transportation systems (ITS) in recent years. Accurate traffic flow prediction which helps to optimize traffic resources and make decisions is an important component of ITS. However, this task is still challenging. Firstly, traffic prediction relies on

the volatility and uncertainty in the traffic flow in the temporal dimension. Secondly, there are very complicated dependencies in the spatial dimension among traffic network nodes and vehicles. The staggered temporal and spatial dependencies make it very difficult to exploit the spatial–temporal dependencies in the traffic flow.

The traffic flow can be considered as a sequence of data continuously recorded by the deployed sensors for a fixed duration. Initially, traffic flow prediction was performed using time series analysis-based methods such as historical average (HA), autoregressive integrated moving average (ARIMA) [1], and vector autoregressive (VAR) [2]. These methods just took the temporal dependencies in traffic flow into account while ignoring the spatial correlations. This defect makes these time series analysis-based models perform less well in practice. To acquire the spatial–temporal

✉ Yun Song
sonie@126.com

¹ School of International College of Engineering, Changsha University of Science and Technology, Changsha 410114, China

² School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

³ Hubei Wujing Freeway Development Company, Wuhan 430020, China

dependencies in traffic flow, traditional machine learning-based methods are applied to this research area, such as support vector regression (SVR) [3] and k-nearest neighbor (KNN) [4]. These traditional machine learning-based methods generally can achieve better prediction than time series analysis-based methods. However, traditional machine learning-based methods can not exploit the spatial–temporal correlations in high-dimensional traffic data sufficiently. In addition, the prediction accuracy of such methods relies heavily on expertise and experience in the field.

Recently, deep learning techniques have been introduced to traffic flow prediction tasks to exploit the implicit spatial–temporal correlations in traffic flow. Convolutional neural networks (CNN)-based prediction models [5–7] can exploit the spatial–temporal correlation in traffic flow. However, their rasterization in the spatial dimension destroys the real spatial structure, which results in the inability to learn the complete spatial dependencies. From the mathematics perspective, the road network is a typical kind of graph-structured data. Graph convolutional neural networks (GCN) pose a natural advantage in processing graph-structured data and multi-graph learning is also a hot topic at the moment. Therefore, GCN-based models [8–11] have been proposed for traffic prediction. They typically combine GCNs with recurrent neural networks (RNN) or CNN to model spatial–temporal correlations in traffic flow. These deep fusion frameworks can model the spatial–temporal dependencies in traffic flow and improve prediction significantly.

Yet, there are disadvantages to these GCN-based methods. Firstly, these methods employed GCN to capture the spatial dependencies in the traffic flow. The node in a traffic network aggregates information based on the degree of interconnection of adjacent nodes with itself [10]. Therefore, the weight of the interaction of each traffic node is particularly important. However, the degree of interconnection between nodes in a road network is dynamic. Spatial correlations cannot be modeled based on the geographical distances between isolated traffic nodes. Moreover, it is difficult for CNN or RNN fusion-based models to learn the long-term dependencies in traffic data [12, 13]. Meanwhile, most deep learning models employ parameter-sharing mechanisms that learn shared traffic patterns between nodes only. In fact, traffic nodes have different traffic patterns. Each traffic node is impacted by points of interest (POI) and other surrounding environments in actual traffic networks [14]. Figure 1 shows an example of the evolution of the traffic flow at three different traffic nodes over three days. Node A may be on the road to the suburbs where it has a relatively low traffic flow. The traffic flow at node B is consistently high during the day and it is probably located on the main road connecting two cities. The traffic flow of node C which may be located on a road connecting two industrial areas has a typical morning peak and evening

peak. This example suggests different traffic network environment leads to different traffic patterns.

To address the issues mentioned above, this paper proposes a traffic flow prediction method, called temporal attention recurrent graph convolutional neural network (TARGCN). The proposed method comprises a node-embedded graph convolutional (Emb-GCN) layer to capture spatial dependencies, a gated recurrent unit (GRU) layer to acquire local temporal dependencies, and a temporal attention (TA) layer to exploit global nonlocal temporal correlations.

The proposed TARGCN is structurally more similar to the RNN-based prediction methods. But, it is different from most of the existing methods. First of all, a new temporal attention layer is employed to extend the receptive field of GRU. This enables the proposed model to capture long-term dependencies in the traffic flow. Moreover, the proposed method applies a node embedding matrix plus a predefined adjacency matrix strategy to model the dynamic correlations between traffic nodes. This node embedding strategy facilitates the proposed method to learn specific traffic patterns for each traffic node. Furthermore, the proposed Emb-GCN layer can fit the adjacency matrix between traffic nodes based on the data adaptively. This allows the model to be more advantageous when the predefined adjacency matrix is not provided or an inaccurate predefined adjacency matrix is provided. The highlights and main contributions of this paper are as follows:

- (1) In this paper, an embedding Emb-GCN layer, a series of gated recurrent units, and a TA layer are proposed to be fused in a network for traffic flow prediction. This fusion framework enables the proposed model to exploit both dynamic spatial correlations between traffic nodes and temporal dependencies between time slices sufficiently.
- (2) An Emb-GCN block is proposed to capture the spatial correlation of traffic flow at a fine-grained level. Node embedding matrix and node parameter learning techniques are applied in Emb-GCN to learn the specific traffic pattern of each traffic node. In addition, the spatial feature matrices learned from the predefined adjacency matrix are filtered with a gating mechanism. This makes the model extract the accurate correlations between nodes from the predefined adjacency matrix as much as possible but discards the inaccurate correlations.
- (3) A series of GRU cells are stacked as a GRU layer to extract traffic patterns from the traffic flow of adjacent nodes in the past few time slices, which enables the model to capture both temporal and spatial correlations simultaneously. Furthermore, an attention layer is applied in the temporal dimension to extend the

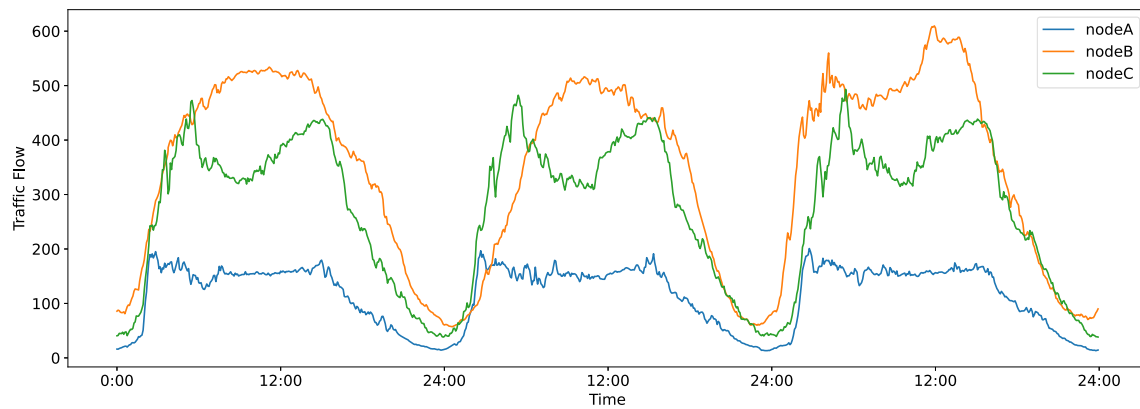


Fig. 1 Traffic nodes have various traffic patterns. The traffic flow level of node A is at a low state for a long time. On the contrary, the traffic flow at node B has been at a high level. Node C has a distinct morning peak and evening peak traffic flow

receptive field of GRU. It facilitates the model to capture not only the local dependencies between adjacent time slices but also long-range nonlocal dependencies between nonadjacent time slices. This strategy enhances the ability of the proposed method for long-time prediction.

- (4) We conducted experiments on three publicly available datasets, PEMS03, PEMS04, and PEMS08. Experimental results verify the effectiveness of the proposed model. The proposed method presents remarkable superiority over state-of-the-art baselines, especially for long-term prediction tasks. Moreover, the implementation of the proposed model is published on Pytorch at <https://github.com/csust-sonie/TARGCN> for further evaluation.

The paper is organized as follows. In Sect. “**Related work**”, the works related to traffic flow prediction are reviewed. In Sect. “**Methodology**”, basic definitions are introduced and the traffic flow prediction problem is formulated first. Then, the proposed TARGCN model is discussed in detail in this section, too. In Sect. “**Experiments**”, the proposed method is evaluated on three public traffic datasets, and ablation experiments are performed. In Sect. “**Conclusion**”, the paper is summarized and some limitations of the proposed method are discussed.

Related work

Traffic prediction is a fundamental and challenging problem for the intelligent transportation system. Traffic prediction methods can be basically divided into time series analysis-based methods, traditional machine learning-based methods, and deep learning-based methods. Time series analysis-based methods treat traffic data as time series data. Williams et al. [1] regressed the lagged values of a smooth time series as well

as the values of the random error term and the lagged values. Lu et al. [2] proposed an extension of the autoregressive (AR) model considering linear correlations between multiple time series. This type of method generally assumes that traffic flow is linear evolution and can only handle smooth sequence data. They do not exploit the internal spatial correlations of traffic data and cannot reflect the non-linearity and uncertainty of traffic data. Traditional machine learning-based methods no longer assume a linear variation in traffic evolution. Wu et al. [3] proposed a regression method for sequence data using the support vector machine. Van Lint et al. [4] proposed a regression model using the k-neighborhood algorithm, which took the weighted average of k-neighbor nodes as the prediction result. These methods can acquire complex dependencies after the addition of manually processed features. They achieved better prediction results than time series analysis-based methods. However, since prediction models are handcrafted, predefined, and fixed, their performance relies on domain knowledge heavily.

In the last few years, deep learning techniques have been introduced to traffic prediction tasks to achieve more accurate predictions. Zhang et al. [6] developed a depth model called ST-ResNet for urban traffic flow prediction based on ResNet [15]. Lin et al. [16] considered point of interest (POI) and crowd movement patterns in the prediction model and applied CNN to acquire spatial-temporal correlations for predicting crowd flows. Yao et al. [17] argued that the periodicity of traffic flow was not strictly periodic but subject to some offsets. For this reason, they proposed a periodicity through the attention mechanism to model the periodicity offset and combined LSTM and CNN to exploit the spatial-temporal correlations. Zhang et al. [18] modeled the flow patterns between edges separately in the urban flow prediction model. Those CNN-based models process the traffic network into rasterized cells to acquire the temporal dependencies to predict future traffic flow. However, rasterization destroys the spatial topology of the road network and fails to exploit the realistic spatial

dependencies of the traffic network. Applying graph convolutional neural networks to exploit the spatial correlation of traffic data can avoid this drawback. Zhao et al. [8] proposed a T-GCN model combining gated recurrent unit (GRU) and graph convolutional neural network (GCN) to capture temporal correlation and spatial correlation, respectively. However, the degree of interconnection between traffic nodes is time-varying, so the static adjacency matrix cannot represent the degree of connection between traffic nodes dynamically. Yu et al. [19] employed two gated convolution layers and a GCN layer to construct an ST-Conv block to capture the spatial-temporal correlation in traffic flow. Geng et al. [20] developed a multi-graph convolutional neural network for traffic prediction employing multiple graphs to encode different features of the traffic network. Defferrard et al. [21] and Kipf et al. [22] applied the spectral graph neural network to exploit the spatial dependencies between traffic nodes. Yanguang et al. [9] defines spatial correlation as a diffusion process and defines the traffic network as a directed graph.

Recently, the attention mechanism has been introduced into traffic prediction for long-term correlation extraction. Guo et al. [11] and Zheng et al. [23] applied attention mechanisms to generate a dynamic adjacency matrix to exploit spatial correlations in traffic flow. The attention mechanism was also applied in the temporal dimension to calculate the degree of association between individual time slices. In addition, traffic data are artificially processed into data with explicit periodicity in [11, 24] designed local spatial-temporal graphs to extract the spatial-temporal dependencies of the traffic data simultaneously. Li et al. [25] improved the local spatial-temporal graph in [24]. It applied dynamic time warping (DTW) to calculate the similarity between time series. However, these methods rely on a predefined static road network adjacency matrix. And their parameter-sharing mechanism makes them only acquire the shared traffic pattern of all nodes. Bai et al. [10] proposed an adaptive parameter learning module with an adaptive graph generation module to learn specific traffic patterns for each node. It utilized a GRU module to acquire temporal correlations, but theoretical and empirical evidence shows that the GRU is difficult to learn to store information for long sequences [26, 27] because of its limited receptive field. Guo et al. [14] employed spectral clustering methods to construct regional micro and macro graphs for the traffic network. The dilated convolution was applied to increase the temporal receptive field in this model, too. However, the spectral clustering method cannot divide the traffic nodes accurately. The ratio of macro and micro graphs needs to be adjusted artificially. The manual adjustment error scheme limits the potential of the model [28].

Most recently, Jiang et al. [29] presented a traffic delay-aware feature transformation prediction model named PDFormer to exploit the time delay in spatial information

propagation. Ji et al. [30] introduced a novel spatio-temporal self-supervised learning framework (ST-SSL) to enhance the representation of traffic patterns. Our previous work [31] proposed the multi-scale spatiotemporal network (MSSTN) to extract and fuse multi-scale spatiotemporal features. Zeng et al. [32] applied the spatial-temporal transformer to capture spatio-temporal correlations using a dynamic graph.

In this paper, a traffic prediction method fusing the node-embedded GCN with a GRU to acquire spatial-temporal correlations is proposed. In addition, a TA layer is employed to extend the receptive field and exploit the global temporal correlations.

Methodology

Problem definition

Traffic Network: In the proposed model, the traffic network is defined as an undirected graph $G = (V, E, A)$, where V denotes the set of nodes in the traffic network. $|V| = N$ denotes the number of nodes, where each node represents a traffic data collection device. E denotes the set of edges in the traffic network, A denotes the adjacency matrix of G .

Traffic Signal Matrix: The traffic signal matrix of the traffic network G at time slice t is denoted as $X_t = (x_{t,1}, x_{t,2}, \dots, x_{t,N}) \in R^{N \times C}$, where $x_{t,i} \in R^C$ denotes the feature vector (a collection of variables of interest) of node i at time t and C denotes the number of features, and the features include traffic flow, speed, and road occupancy.

Traffic Prediction problem formulation: Given a historical spatial-temporal traffic signal matrix $X = (X_1, X_2, \dots, X_h) \in R^{h \times N \times C}$ over the past h time slices and a traffic network graph $G = (V, E, A)$. The goal of traffic prediction is to build a model $f(\cdot)$ that aims to predict the traffic signal matrix $Y_{pre} = (X_{t+1}, X_{t+2}, \dots, X_{t+T'}) \in R^{T' \times N \times C}$ over the next T' time slices.

Attention Mechanism: The attention mechanism is a method to model the dependencies between a collection of values and the target under a query by adaptively assigning to each value in the collection a weight that is determined by the query and keys associated with them.

The self-attention mechanism is a special expression of the attention mechanism. The key, query, and value in the self-attention mechanism are obtained by a linear transformation of the input vector. So self-attention is better at capturing the internal correlation of data. It is defined as (1).

$$Att(Q, K, V) = softmax\left(\frac{W_q Q \cdot (W_k K)^T}{\sqrt{d_m}}\right)V \quad (1)$$

where W_q, W_k are learnable matrices, Q, K, V and d_m are query, key, value, and their dimension respectively.

Architecture of the proposed method

The proposed TARGCN model consists of a node-embedded GCN layer, a series of GRUs, a TA layer, and sequentially a convolutional layer as illustrated in Fig. 2. It employs node embedding to generate a dynamic adjacency matrix in the spatial dimension which models the dynamic associations between nodes adaptively. Following this layer, a series of GRUs is applied to model local temporal correlations in the temporal dimension. Then, global long-range temporal correlations are exploited by a TA layer.

Initially, the original traffic flow matrix at time $t X_t \in R^{N \times d_{in}}$ is merged with the initial hidden state $H_{t-1} \in R^{N \times d_h}$. It is then fed into the graph convolutional neural network layer along with the initialized node embedding matrix $E_{adj} \in R^{N \times d_e}$ to obtain the spatial correlation matrix at time $t X'_t \in R^{N \times (d_{in}+d_h)}$. After that, the hidden state is updated by passing $X'_t \in R^{N \times (d_{in}+d_h)}$ through the reset gate and the updating gate in GRU. Then, $X'_t \in R^{N \times (d_{in}+d_h)}$ of T_h time slices are concatenated as $X' \in R^{N \times (d_{in}+d_h) \times T_h}$, and temporal position embedding is added for X' . Finally, the concatenated spatial correlation matrix X' exploits the global temporal correlations through the TA layer and the final prediction values are acquired by the 1D convolution layer. Where N denotes the number of nodes, d_{in}, d_h , and d_e denote the input dimension, hidden dimension, and node embedding dimension, respectively. And T_h is set to 12 in this work to predict the traffic flow in the next hour.

Spatial correlation modeling

The proposed model employs a spatial domain graph convolutional neural network to capture spatial correlations at a fine granularity. It can be defined as (2).

$$GCN(A, X) = \sigma \left(I_N + D^{-\frac{1}{2}} A D^{-\frac{1}{2}} \right) X W + b \quad (2)$$

where $A \in R^{N \times N}$ is the adjacent matrix of the traffic graph, D is the degree matrix of A . $X \in R^{N \times C}$ is the traffic signal matrix. $W \in R^{C \times c'}$ and $b \in R^{c'}$ are learnable parameters. σ denotes a nonlinear activation function.

For the adjacent matrix, existing works mainly utilize the geographic distances between traffic nodes or similarity functions to model it dynamically. However, the predefined adjacency matrix is normally calculated from the geographical distance between traffic nodes, but the node association expressed by this predefined adjacency matrix is inaccurate. Figure 3 is a schematic diagram of a section of the road network structure, the orange points represent different traffic

nodes. The red gradient area represents the degree of association between node v_1 and other nodes in the predefined adjacency matrix, the darker the red means the association is stronger. It can be seen from Fig. 3 that v_1 is more closely associated with v_5 than v_1 is with v_2 . But in reality, v_1 and v_2 are more closely correlated, and even v_1 and v_4 , which are very far apart, are more strongly correlated than v_1 and v_5 . Therefore, the predefined adjacency matrix obtained by geographic distance calculation alone is inaccurate, but the manual design of the predefined adjacency matrix requires a specific design for each region with the expertise of the relevant domain and lacks generalization performance. Just calculating the similarity between sequence data to represent the adjacency matrix generally cannot capture the spatial correlations fully, which may result in considerable biases. So, the proposed model applies a learnable node embedding matrix $E_{adj} \in R^{N \times d_e}$ to infer hidden interconnection from the traffic flow automatically. Each row of E_{adj} represents the embedding vector of one traffic node. Then, the spatial dependencies between every individual node are inferred by multiplying E_{adj} and E_{adj}^T , which is the transpose matrix of E_{adj} . Where N denotes the number of nodes and d_e represents the embedding dimension of E_{adj} . In addition, $D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ is calculated as a whole as (3) to reduce computational resources.

$$\tilde{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}} = \text{softmax} \left(\text{Relu} \left(E_{adj} \cdot E_{adj}^T \right) \right) \quad (3)$$

where $\text{softmax}(\cdot)$ and $\text{Relu}(\cdot)$ represent nonlinear activation functions, $\tilde{A} \in R^{N \times N}$ is applied to the graph convolution operation. E_{adj} will be updated continuously during training automatically. Then, the graph convolution operation is formulated as (4).

$$GCN(\tilde{A}, X) = (I_N + \tilde{A}) X W + b \quad (4)$$

From another perspective, since the predefined adjacency matrix generally contains prior experiences, it can still reflect the inherent correlation between certain traffic nodes. Therefore, the predefined adjacency matrix should not be discarded in the prediction model completely. In ICGRRN [33], the predefined adjacency matrix is added to the dynamic adjacency matrix of the embedding matrix fit as the parameter of the graph convolution operation. Different from ICGRRN [33], the proposed TARGCN puts the predefined adjacency matrix and the dynamic adjacency matrix into separate graph convolution operations. The result of the predefined adjacency matrix is further processed using a gating mechanism and a linear layer. Then, it is summed with the learned embedding matrix. This process reduces the impact of inaccurate data in the predefined adjacency matrix.

As shown in (5), a parameter α is set in the proposed model to control the weight of the predefined adjacency matrix.

Fig. 2 TARGCN architecture. A series of GRUs embedded with GCN is applied to exploit spatial and local temporal correlations. And, a TA layer is employed to acquire the global temporal correlations. Moreover, a residual connection is applied between the temporal position layer and the TA layer

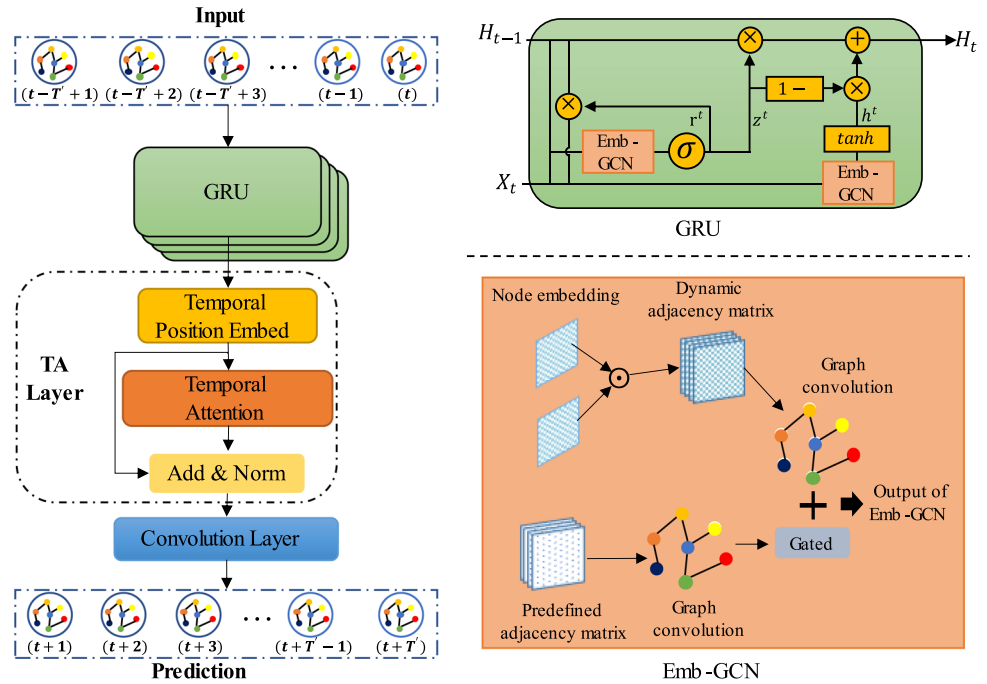
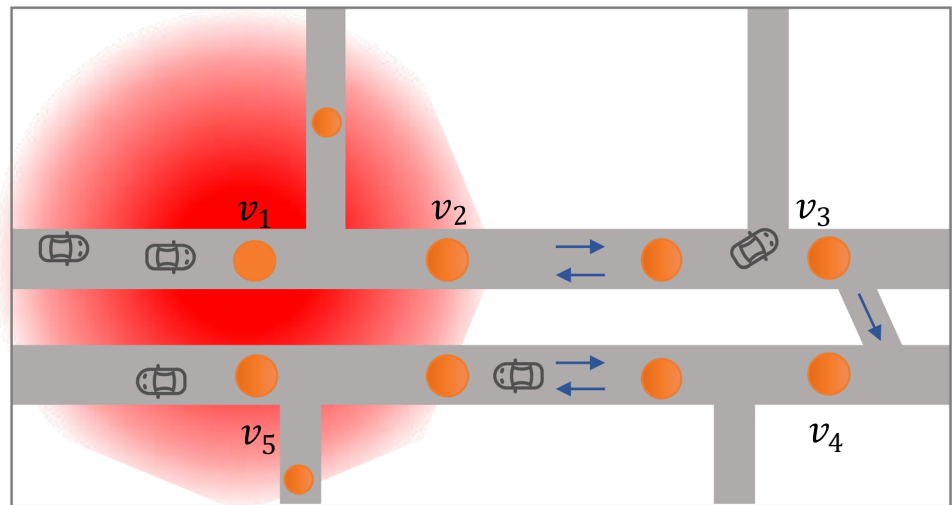


Fig. 3 Road network structure diagram. The degree of correlation between nodes in the road network that are geographically nearby is not always strong



When $\alpha = 0$, it means no predefined adjacency matrix is applied. Specifically, a gate mechanism is applied in the proposed model to enable Emb-GCN to discard the insignificant information in the predefined adjacency matrix.

$$GCN_{sd}(\tilde{A}, X) = \alpha * sigmoid(X_s) \odot X_s + GCN(\tilde{A}, X) \tag{5}$$

$$X_s = W_s \cdot GCN(softmax(A^{pre}), X) + b_s \tag{6}$$

$$A_{i,j}^{pre} = \begin{cases} 0, & distance(i, j) > \frac{1}{\tau} \\ 1/distance(i, j), & distance(i, j) \leq \frac{1}{\tau} \end{cases} \tag{7}$$

where $distance(i, j)$ is the distance between node i and node j . $\tau \in (0, 0.1)$ is an artificial threshold. When the distance between node i and node j is greater than $1/\tau$, these two nodes are considered to be uncorrelated with each other. And W_s and b_s are learnable parameters.

The GCN operation can be regarded as aggregating the features of neighbor nodes into the central node, where all nodes share the parameter matrix W and b . This parameter-sharing mechanism makes the model can only learn similar traffic patterns for all nodes. In reality, nodes of different geographical locations in the traffic network will be influenced by their surrounding environment differently. Adjacent node sets or even each node poses its own specific traffic mode. So, a specific parameter is given for each node in this

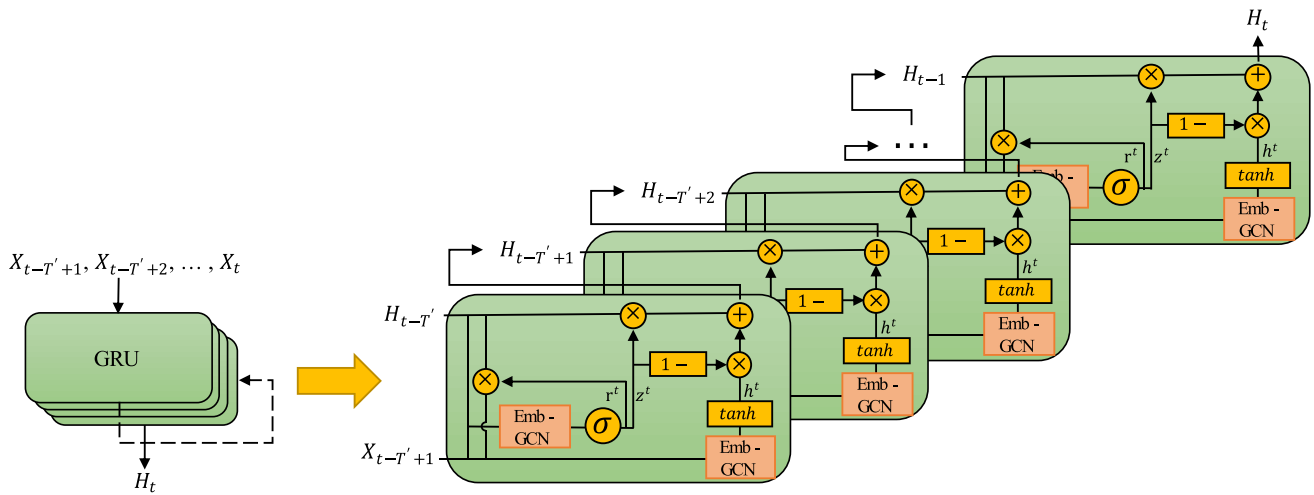


Fig. 4 GRU structure diagram, TARGCN employs multiple GRUs connected in series to capture the local temporal correlation of traffic flow. The left part illustrates the overall structure of the GRU layer, and the implementation details between individual GRU units are shown on the right

work. This forces each node in the traffic network to learn a specific traffic pattern. Specifically, two learnable matrices $W_E \in \mathbb{R}^{N \times d_{in} \times d_{out}}$ and $B_E \in \mathbb{R}^{N \times d_{out}}$ are embedded in the model to assign parameters to each node. However, when N is large, W_E and B_E will also become very large. To reduce the number of parameters, two smaller matrices $w_e \in \mathbb{R}^{d_w \times d_{in} \times d_{out}}$ and $b_e \in \mathbb{R}^{d_w \times d_{out}}$ are employed in the proposed model to construct W_E and B_E , where d_w is the embedding dimension of the same size as in the node embedding matrix E_{adj} , and $d_w \ll N$. Each row of W_E and B_E represents a specific weight and bias of a node, respectively. They can be expressed as (8) and (9).

$$W_E = E_{adj} \cdot w_e, B_E = E_{adj} \cdot b_e \tag{8}$$

$$GCN(\tilde{A}, X) = (I_N + \tilde{A})XW_E + B_E \tag{9}$$

Temporal correlation modeling

Compared with RNN, GRU enhances the ability for sequence-dependent modeling further. The proposed model considers employing multiple stacked GRUs to exploit local temporal correlations in the temporal dimension. Each GRU cell blends the flow of the corresponding time slice and the output state of its previous cell to extract temporal features. Figure 4. shows the overall structure of the proposed GRU layer and the implementation details between the GRU cells. Given the traffic flows of the last T' time slices to predict the flow of the t -th time slice, we stack T' GRU cells as a GRU layer. Firstly, a hidden state $H_{t-T'}$ is initialized and then pushed into the first GRU cell with the traffic flow of the time slice $t - T' + 1$, that is $X_{t-T'+1}$. Then, the output of this cell $H_{t-T'+1}$ and the traffic flow of the second

time slice $X_{t-T'+2}$ are input into the second GRU cell, and so on. Sequentially, the last cell outputs the state H_t as the temporal features extracting by the whole GRU layer. Specifically, each GRU cell can be formulated as (10). Note that, we replace the standard matrix multiplication in GRU with the graph convolution operation in (10). This operation enables GRU to acquire both spatial correlation and local temporal correlation. The number of historical time slices applied for prediction T' is set to 64 in this work, just as other state-of-the-art methods. So, there are 64 GRU cells in the GRU layer.

$$z^t = \sigma(GCN_{sd}(\tilde{A}, X_t || H_{t-1})),$$

$$r^t = \sigma(GCN_{sd}(\tilde{A}, X_t || H_{t-1})),$$

$$h^t = \tanh(GCN_{sd}(\tilde{A}, X_t || (r^t \odot H_{t-1}))),$$

$$H_t = z^t \odot H_{t-1} + (1 - z^t) \odot h^t. \tag{10}$$

where $||$ denotes the union operation, X_t and H_t denote the traffic flow and hidden state at time t respectively, \odot represents the Hadamard product, $\sigma(\cdot)$ denotes the *sigmoid* activation function. z^t and r^t are reset gate and update gate at time t , respectively.

After the GRU layer, a self-attention module as shown in Fig. 5 is followed to capture the correlation degree between individual time slices in the time dimension in the proposed method. This structure extends the receptive field and makes up the deficiency of GRU in exploiting long time series temporal correlations, In this module, the temporal position embedding is first added for the input. And then multiple temporal attention operations are performed with

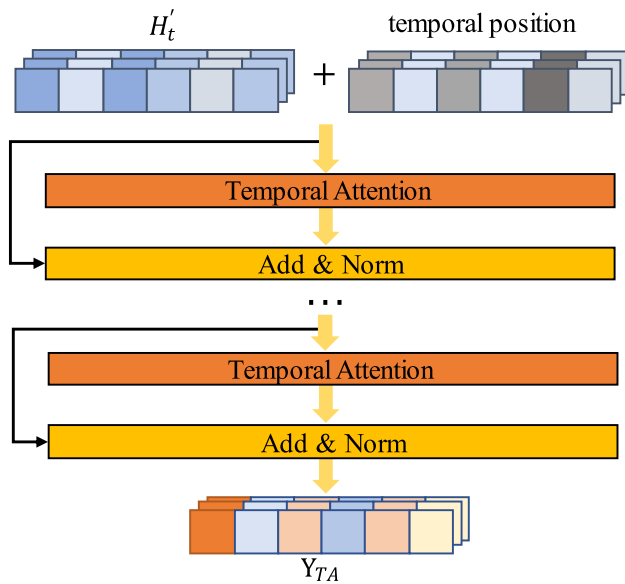


Fig. 5 The structure of the TA layer

each layer connected with residuals. Note that, we calculate the attention scores in the temporal dimension after transforming the output of GRU to query, key, and value by convolution operations. Different from the traditional self-attention mechanism, we substitute the linear transformation on the query and key with 1D convolution here. Since the convolution operation can acquire semantic information in the context, this substitution enables the proposed model to exploit the global nonlocal temporal correlation between nonadjacent time slices. The attention score is defined as follows.

$$Att(Q, K, V) = softmax\left(\frac{Q_c(Q) \cdot K_c(K)^T}{\sqrt{d}}\right) V_l(V) \quad (11)$$

$$Q_c(Q) = Conv(Q) \quad (12)$$

$$K_c(K) = Conv(K)$$

$$V_l(V) = Linear(V)$$

where Q , K , V , and d are the query, key, value, and their dimensions, respectively. $Conv(\cdot)$ and $Linear(\cdot)$ denote convolution operation and linear transformation, respectively. $Q_c(Q)$ and $K_c(K)$ represent the query vector and key vector after the convolution operation, respectively, $V_l(V)$ is a vector of values that have been linearly transformed. d denotes the features dimension of traffic flow. In the convolution operation, the convolution kernel size is set to 3 in this work.

Since the attention function treats each position equally in the self-attention mechanism, the sequential dependencies

in the traffic flow are normally discarded in the calculation process [13]. However, the order information is crucial to the task of modeling time-series data because there is a stronger correlation between data at closer distances. For example, the traffic flow at 9:00 am is more instrumental for predicting the traffic flow at 9:30 am than that at 8:00 am. To address this issue, we append the temporal position code from the original data as (13). Specifically, we append a position code e_{tp} for the time slice t as (14) to identify the position of $H_t[i, :]$ for i -th feature in the whole sequence H_t . The TA layer can be expressed as (15).

$$H_t'[i, :] = H_t[i, :] + e_{tp} \quad (13)$$

$$e_{tp} = \begin{cases} \sin(t/(1000)^{2i/d}), & t = 0, 2, 4, \dots \\ \cos(t/(1000)^{2i/d}), & t = 1, 3, 5, \dots \end{cases} \quad (14)$$

$$Y_{TA} = Relu(Linear(Att(H'_t, H'_t, H'_t))) + H'_t \quad (15)$$

where $\sin(\cdot)$, $\cos(\cdot)$, $Relu(\cdot)$ represent sine, cosine, and the rectified linear unit function, respectively. d is the number of features.

Following the TA layer, a 1D convolutional module is employed as the prediction layer. Note that, to reduce the error accumulation caused by predicting one time slice each time, the prediction layer is designed to infer the predicted values for the next 12 time slices at one time in the proposed framework.

In summary, the proposed model includes three main modules, an Emb-GCN layer, a GRU layer, and a TA layer. In Emb-GCN, the adjacency matrix of the traffic network is dynamically modeled by an embedding matrix. Meanwhile, the parameter learning matrices are applied to learn specific traffic patterns for each node. In the GRU layer, a graph convolution operation is utilized instead of matrix multiplication to exploit spatial correlations and local temporal correlations. Then, a TA layer with a temporal location embedding module is employed to exploit global temporal correlation and a 1D convolutional layer is utilized as a prediction layer. The improved GRU and TA layer enables the model to effectively exploit the spatial-temporal correlation in traffic flow. In addition, a 1D convolutional layer for one-time inference of prediction results avoids error accumulation.

Experiments

Datasets

The proposed model is evaluated on three public traffic datasets PEMS03, PEMS04, and PEMS08 as shown in Table 1. All of them are real freeway traffic flows in California collected by the Caltrans Performance Measurement

Table 1 Experiment datasets

Datasets	Nodes	Samples	Time range
PEMS03	358	26,208	01/09/2018–30/11/2018
PEMSD4	307	16,992	01/01/2018–28/02/2018
PEMSD8	170	17,856	01/07/2016–31/08/2016

System (PEMS) [34] in real-time every 30 s. The traffic data is aggregated into 5 min, which means 288 traffic data points per day. The Nodes field in Table 1 represents the number of nodes in the dataset, where each node represents a sensor used to collect traffic data in the real world. The Samples field in Table 1 indicates the number of data points collected at each node, and the Time Range field indicates the period over which the dataset was collected.

- (1) PEMS03: This dataset contains traffic flow data from 358 traffic collection nodes for the months 01/09/2018–30/11/2018. Note that, there are some very small values in the PEMS03 dataset. They may cause the MAPE indicator to be abnormally large. To address this issue, the flow rate values less than 10 are set to 0 in the experiments.
- (2) PEMS04: This dataset contains traffic flow data from 307 traffic collection nodes for the months 01/01/2018–28/02/2018.
- (3) PEMS08: This dataset contains traffic flow data from 170 traffic collection nodes for two months 01/07/2016–31/08/2016.

The traffic networks in these two datasets are defined as undirected graphs. The traffic data contains three characteristics: traffic flow, speed, and road occupancy. In the experiment, traffic flow is the prediction target while traffic speed and road occupancy will not be involved in the model training. The traffic flow data set is divided into training sets, validation sets, and test sets according to 6:2:2. The target of the proposed model is to predict the traffic flow for the next 12 time slices using the historical traffic flow in the last 12 time slices.

Baseline methods

The proposed TARGCN model is compared with the following baseline methods:

- (1) VAR [2]: Vector autoregression is a time series model that portrays pairwise correlations between multiple time series.
- (2) DCRNN [9]: The diffusion convolutional recurrent neural network employs diffusion graph convolutional

networks and GRU based on seq2seq to predict traffic graph series data.

- (3) STGCN [19]: This method employs ChebNet in the spatial dimension and 2D convolutional networks in the temporal dimension to model the correlations in spatial–temporal graph data.
- (4) ASTGCN [11]: This is an attention-based spatial–temporal graph convolutional network which employs spatial attention and temporal attention mechanisms to model spatial–temporal dependencies.
- (5) STSGCN [24]: This spatial–temporal synchronous graph convolutional network proposes a local spatial–temporal graph to model spatial–temporal correlations.
- (6) AGCRN [10]: The adaptive graph convolutional recurrent network utilizes a GCN with data-adaptive graph generation in the spatial dimension. In this model, the temporal and spatial correlations are captured by embedding the GCN into the GRU.
- (7) STGMN [35]: A 1D-CNN based on channel attention mechanism and “inception” structure is proposed to extract temporal correlation. An interpretable multi-graph gated graph convolution framework is proposed to extract the spatial correlation.
- (8) IGCRN [33]: An improved Graph Convolution Recurrent Network for spatial–temporal dependence capturing and traffic flow prediction.
- (9) MAGRN [36]: A multi-scale attention-based graph convolutional recurrent network framework for multi-scale feature extraction and dual attention mechanisms for traffic flow prediction.
- (10) SLTTCN [37]: A convolution network model that employs spatial linear transformers to aggregate spatial information and bidirectional temporal convolution networks to capture temporal dependencies in traffic flow g.

Experimental settings

In the experiments, the MAE, RMSE, and MAPE are employed as evaluation metrics. They are defined as (16)–(18).

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (16)$$

$$RMSE = \left(\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 \right)^{1/2} \quad (17)$$

$$MAPE = \frac{100\%}{N} \sum_{i=1}^N |(Y_i - \hat{Y}_i)/Y_i| \quad (18)$$

Table 2 Hyperparameter settings in TARGCN

Hyperparameter	Node Embedding Dimension	Number of GRUs	Number of TA layers	Batch size	epoch	Initial learning rate	α
PEMS03	10	64	2	64	150	0.003	1
PEMSD4	10	64	2	64	200	0.003	1
PEMSD8	2	64	2	64	300	0.003	1

where N denotes the number of samples, and Y_i and \hat{Y}_i denote the ground truth and the prediction of the i -th sample, respectively. Note that, all these three metrics are prediction error evaluation indicators. Smaller values mean better predictions.

The proposed model is implemented on the PyTorch 1.9.1 deep learning framework with one NVIDIA Tesla P100-PCIE 16GB card. The mean absolute error is utilized as the loss function. Adam is employed as the optimizer. The learning rate is initialized to 0.003 and decayed during training. The detailed hyperparameter settings are listed in Table 2. All the flow data are normalized by employing standard normalization methods first before they are input to the neural network during the training phase. And during the inference phase, the predicted values are recovered to the actual flows by the mean and standard deviation of the flow recorded in advance.

Note that, the node embedding dimensions are selected based on the number of nodes in the dataset. They are selected in [2, 4, 6, 8] for PEMS03 and PEMS04 with less number of nodes, and [8, 10, 12, 16] for PEMS03 and PEMS04 with more number of nodes. The number of GRUs, the number of TA layers, the batch size, and The initial learning rate are chosen in [32, 64, 128], [1, 2, 3, 4], [16, 32, 64], and [0.01, 0.005, 0.003, 0.001], respectively. A learning rate decay strategy is set up so that the learning rate decreases as the epoch increases.

Experiment results

Table 3 lists average prediction errors over 12 prediction steps in the next hour in terms of various metrics for all compared methods. All results of the proposed method are averaged over three experiments. Results of other methods come from their papers unless they open their source codes. It should be noted that the prediction errors of IGCRN [33] on the PEMS03 are omitted from the table because there are neither source codes publicly available for it nor experimental results on the PEMS03 dataset in the original paper. As shown in the table, the VAR [2] model considers spatial-temporal correlations among multiple time series, but it is representation ability to model dynamic spatial-temporal correlations is weak. Thus, the prediction performance of this method is limited.

The deep learning-based methods present an enormous advantage over the traditional methods. This demonstrates the superiority of deep learning techniques in extracting non-linear and dynamic dependencies of traffic flow. The deep learning-based methods ASTGCN and STSGCN applied 1D-CNN and GCN to exploit temporal and spatial correlation. It is difficult to exploit long-term temporal correlation due to the limitation of CNN by the size of the convolutional kernel. STGCN employs TCN to expand its receptive field in the temporal dimension. But TCN still requires a stack of $O(\log_k(T_h))$ convolutional layers to connect any two positions in the sequence, where k is the convolution kernel size [13]. Therefore, it is still difficult for TCN to exploit the long-term temporal correlation of traffic data [12]. DCRNN and AGCRN are both traffic prediction models based on GRU and GCN. The parameter-sharing mechanism makes DCRNN learn only the shared traffic patterns of all nodes, resulting in a much higher prediction error than the proposed TARGCN, especially for long-term predictions. STGMN applied a channel-focused multi-resolution CNN and an interpretable multi-graph framework to exploit temporal correlation and spatial correlation, respectively. IGCRN employs an embedding matrix combined with a predefined adjacency matrix to fit the dynamic adjacency matrix in the spatial dimension and an LSTM module to capture temporal correlation. The performance improvement of TARGCN over IGCRN may primarily result from the construction method of the embedding matrix and the TA layer. The matrix embedding of TARGCN allows for more fine-grained learning of traffic patterns of traffic nodes, and the TA layer further captures the global temporal correlation with attention mechanisms.

As illustrated in Table 3, the proposed TARGCN achieves the best average prediction among all compared methods both on PEMS03 and PEMS04. On PEMS04, the proposed method is slightly inferior to MAGRN and SLTTCN in terms of RMSE and IGCRN in terms of MAPE. Still, it achieves the best MAE among all methods. These results demonstrate the advantages over other state-of-the-art traffic prediction methods. Prediction improvements mainly stem from more efficient extraction of spatial-temporal features in the traffic flow. In the spatial dimension, TARGCN applies the node embedding matrix and node parameter learning into

Table 3 Comparison on average prediction errors over 12 steps

Baseline methods		VAR	DCRNN	STGCN	ASTGCN(τ)	STSGCN	AGCRN	STGMN	IGCRRN	MAGRNN	SLTTCN	TARGCN (proposed)
Datasets	Metrics											
PEMS03	MAE	21.08	18.39	18.28	17.85	17.51	16.24	17.21	-	-	-	15.49
	RMSE	34.75	30.56	30.73	29.88	29.05	28.72	29.85	-	-	-	27.06
	MAPE (%)	22.35	20.22	17.52	17.65	16.92	13.07	15.49	-	-	-	12.20
PEMSD4	MAE	23.75	23.64	22.09	22.31	20.76	19.83	21.06	20.13	19.29	19.69	19.22
	RMSE	36.62	36.84	34.83	34.65	33.78	32.28	32.61	32.20	31.31	31.45	32.08
	MAPE (%)	18.07	16.03	14.24	15.87	13.75	12.93	14.23	12.21	12.71	-	12.76
PEMSD8	MAE	22.32	18.21	17.85	18.81	17.11	16.13	15.95	15.58	16.23	16.74	15.14
	RMSE	33.83	28.21	29.56	28.41	26.76	25.46	24.69	24.87	26.34	25.87	24.62
	MAPE (%)	14.47	11.58	11.06	12.51	10.91	10.19	11.08	10.12	10.90	-	9.90

the graph convolutional network to construct the Emb-GCN module for exploiting spatial correlations. In the Emb-GCN module, a gate mechanism is designed to filter the inaccurate parts of the predefined adjacency matrix to reduce the perturbation of the model caused by inaccurate data. The Emb-GCN module enables TARGCN to fit the traffic adjacency matrix well and acquire specific traffic patterns for each node. The GRU exploits temporal correlation in the traffic flow by updating the hidden states continuously in the temporal dimension. It calculates the temporal correlation between two distant time slices requiring all the time slices between them. However, in this process, the gate mechanism makes GRU discard certain information resulting in incomplete long-term temporal correlation. Therefore, with the increase of the prediction interval, the ability of GRU to capture temporal correlation becomes weaker. In contrast, the TA layer first adds a temporal location embedding to the traffic flow recording the order information of the data. Then, it employs the attention mechanism to calculate the correlation degree between individual time slices. Employing the attention mechanism to calculate the degree of intercorrelation between two slices is not affected by the other time slices, so the captured long-term temporal correlation is more complete. The proposed TARGCN captures spatial-temporal correlation by GRU embedded with GCN. Moreover, the model employs a TA layer to capture the global temporal correlation, which exploits the spatial-temporal correlation of traffic flow more adequately than other methods do.

Figure 6 illustrates curves in terms of error metrics for all methods for 12 prediction intervals. As seen in Fig. 6, all the machine learning methods, including SVR and deep learning-based methods achieve satisfactory prediction for the 5-min prediction task. For short-term intervals, especially ultra-short-term intervals, traffic flow is highly temporal-dependent. Therefore, methods exploiting temporal dependencies in traffic flow, even time series analysis-based methods can make good predictions. However, as the prediction interval increases, nonlinearity and uncertainty are growing. Data-driven methods become more advantageous than fixed model-based methods. Meanwhile, the traffic node is increasingly impacted by its adjacent nodes. Spatial correlations play a more and more important role in traffic evolution. From ten-minute prediction, methods begin to exhibit differentiation in prediction accuracy. Deep learning-based methods, especially methods taking both temporal and spatial correlations into account start to stand out. The proposed method utilizes a graph convolution operation in GRU to exploit spatial correlations and local temporal correlations. Besides, a TA layer is employed to extract long-term temporal correlations. It naturally makes more competitive prediction than other methods. It outperforms other methods in terms of most metrics for ten-minute prediction except slightly behind AGCRN in terms of RMSE on PEMS03.

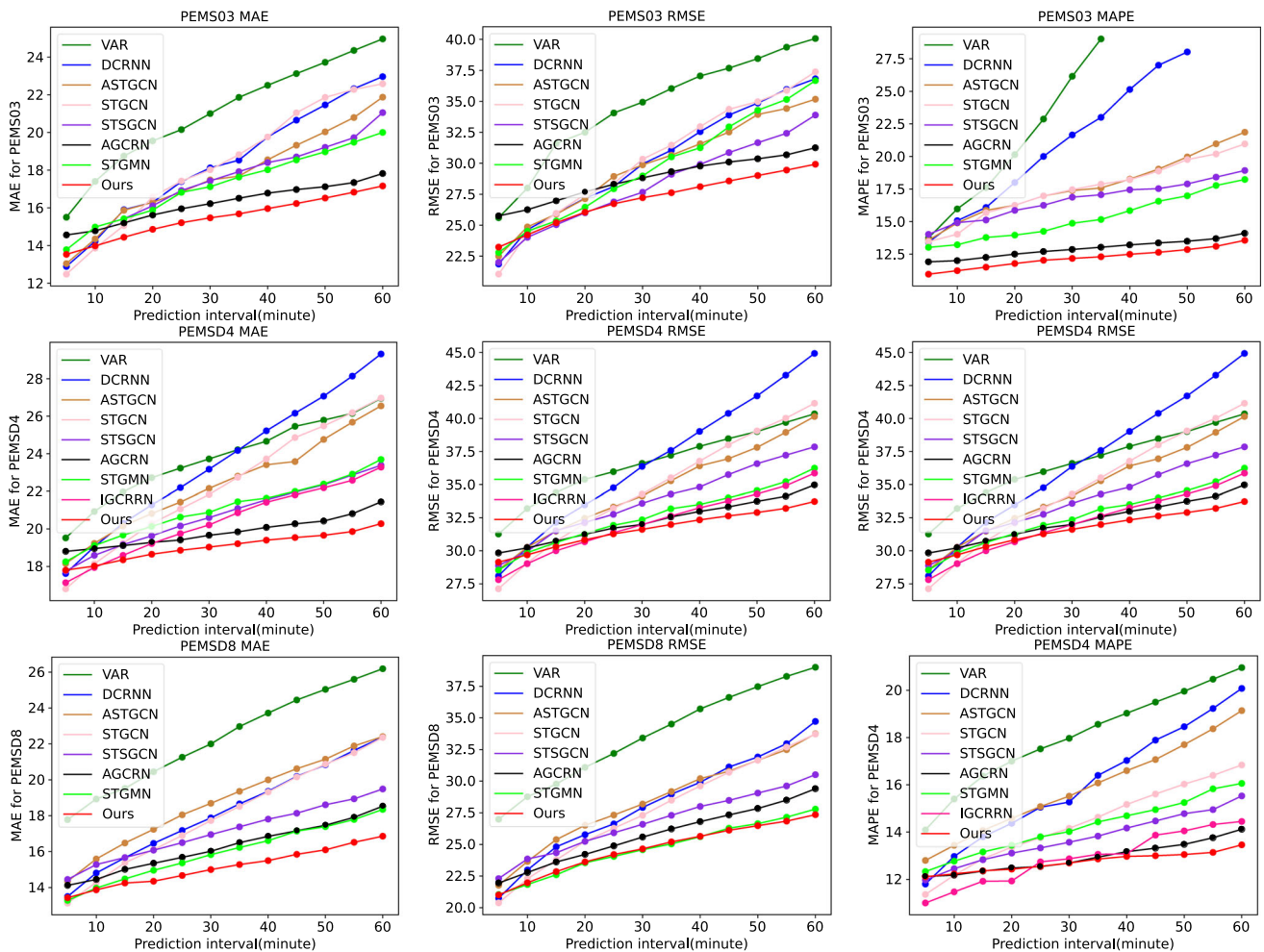


Fig. 6 Prediction error curves for various intervals on PEMS04 and PEMS08

For 30-min and longer predictions, it exceeds all baseline methods in terms of all metrics, and the superiority is significant over most methods. Only the gap between AGCRN and the proposed model is smaller in terms of some metrics. Moreover, it can be observed that the curves of the proposed TARGCN are gentler than compared methods. In general, the longer the prediction intervals, the more difficult the prediction task is. The prediction accuracy of all models degrades with the interval increase. It can be observed from Fig. 6 that the proposed model performs less prediction accuracy degradation than other state-of-the-art baselines with the interval increase. This demonstrates that the proposed method takes full advantage of the spatial correlation of the road network and further models the long-term temporal correlation with a temporal self-attention mechanism, which leads to a more accurate result for long-term prediction.

Figure 7 shows the comparison of predicted values and actual values for four traffic nodes over four days in the 60-min prediction task. The prediction details are illustrated more intuitively. As presented in Fig. 7, the traffic of all

nodes presents obvious periodicity by day. However, there exist large differences in the traffic patterns for nodes and periods. There are two traffic peaks for the 115th node of PEMS03. One is between 4:00 am and 5:00 am. Another one comes at about noon. However, the other two nodes just have one traffic peak. Moreover, the peaks for these two nodes are different in shape. Even traffic peaks of the 127th node of PEMS08 are totally different in shape between day and day as shown in Fig. 7c. Despite these challenges, it can be observed that the proposed TARGCN simulates the real traffic flow of nodes with different traffic patterns evolution very well. Compared with the state-of-the-art prediction method AGCRN [10], its prediction curves are closer to the ground truth in most cases. Even there occur occasionally instantaneous dramatic changes without any portent in the actual traffic. The predictions do not deviate from the real trends. This demonstrates the robustness of the proposed method to the noise flow in the traffic. For some aperiodic flow changes lasting a longer duration, the proposed method also makes a good prediction. That is, the proposed method captures

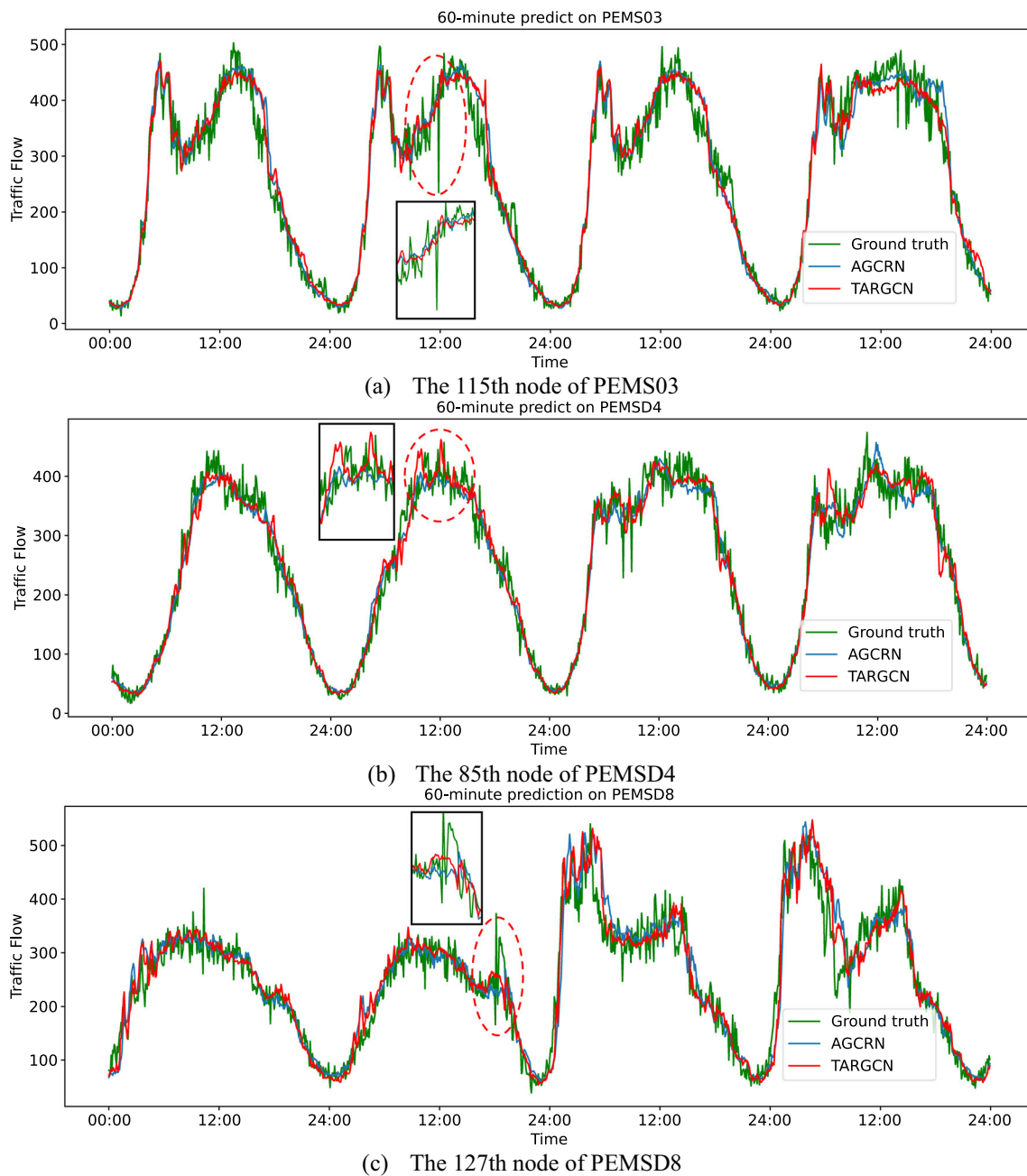


Fig. 7 Visualization of 60-min predictions on PEMS03, PEMS04, and PEMS08

the traffic changes of the connected nodes for the predicting node and makes correct inferences. This shows the proposed method extracts the spatial-temporal features in the road network effectively.

Ablation experiments

Ablation experiments are performed on the PEMS08 dataset to verify the effectiveness of each module in TARGCN.

- (1) TARGCN-linear: A linear layer is applied to replace the learning parameter matrices w_e and b_e for each node in this ablation model. This experiment is to verify the inadequacy of the parameter-sharing mechanism in the traffic prediction task.
- (2) TARGCN-SA: This ablation model applies an attention mechanism combined with a static adjacency matrix to generate the dynamic adjacency matrix. It is to investigate the effectiveness of the node embedding in TARGCN.

- (3) TARGCN-noTA: This model removes the TA layer from the original TARGCN model. It is to verify the effectiveness of the TA layer.
- (4) TARGCN-noGate: This experiment removes the gating mechanism from the Emb-GCN. The purpose of this controlled experiment is to verify that the gating mechanism effectively utilizes the predefined adjacency matrix to obtain better prediction performance of the proposed model.

The results of the ablation experiments are illustrated in Table 4 and Fig. 8. As shown in Table 4 and Fig. 8, TARGCN achieves the best prediction, then followed by TARGCN-noGate, TARGCN-noTA, TARGCN-SA, and TARGCN-linear. The TARGCN-linear model employs a parameter-sharing mechanism that makes the model learn shared traffic patterns for all traffic nodes. This parameter-sharing mechanism results in most nodes not being assigned accurate parameters and the limited prediction accuracy of the model. By contrast, TARGCN and TARGCN-noTA learn specific parameters for each node, so they make much better predictions than TARGCN-linear.

TARGCN-SA combines the self-attention mechanism with a predefined adjacency matrix to capture spatial correlations. First of all, the single self-attention mechanism cannot extract spatial correlations fully. Secondly, since the predefined adjacency matrix is obtained by calculating the static geographical distance between nodes, it does not represent degrees of dynamic correlations between nodes accurately. In the training process, the inaccurate predefined adjacency matrix repeatedly participates in the calculation, which leads to accumulative errors. In TARGCN and TARGCN-noTA, the learnable node embedding matrices are employed to model dynamic adjacency matrices, which benefits prediction.

As illustrated in Table 4 and Fig. 8, predictions of sTARGCN are better than those of TARGCN-noGate. This validates the effectiveness of the gating mechanism. Specifically, the data in the predefined adjacency matrix that differ significantly from the actual correlations have a negative impact. The gating mechanism can effectively reduce the negative effects of such inaccurate correlations, so that the TARGCN can utilize the accurate information in the predefined adjacency matrix without being severely perturbed, thus achieving the best prediction performance.

To more intuitively evaluate the performance of methods, predicted traffic flows at the 127th node of PEMS8 for TARGCN and TARGCN-noTA are presented in Fig. 9. It can be seen from Fig. 9 that for TARGCN, although the prediction deviation increases slightly with the increase of the prediction interval, the predicted value is always well attached to the ground truth. The red circle area in Fig. 9 shows that there exists a relatively large margin from the ground truth for

TARGCN-noTA. And the prediction error enlarges with the increase of the prediction interval more significantly than TARGCN does. This verifies that the TA layer employed in the proposed method remedies the disadvantage of GRU for long-term prediction and guarantees the prediction is steady.

Complexity

To evaluate the efficiency and complexity of the proposed model, an experimental assessment is conducted on the parameters of all comparative baselines and the TARGCN model. Table 5 presents the parameter quantities for TARGCN and other state-of-the-art deep-learning traffic prediction methods. As shown in Table 5, all models except STSGCN have fewer than one million parameters. STGCN and DCRNN have relatively fewer parameters, each with less than 300,000. This is because both models leverage graph convolutional networks to capture spatial correlations and combine them with temporal convolutional networks and gated recurrent units to capture temporal correlations. Their relatively simple architectures are suitable for light traffic networks with fewer nodes. STSGCN, on the other hand, significantly increases the parameter count to 2 million due to its spatial-temporal synchronous graph convolution layer, which incorporates neighboring spatial-temporal features to construct a spatial-temporal adjacency matrix. Despite achieving high prediction accuracy, its complexity and efficiency are considerably affected. The proposed TARGCN, along with baseline methods ASTGCN, AGCRN, and STGMN, all have parameter counts below one million. These models are well-suited for complex traffic networks and can meet real-time prediction requirements. Most existing deep learning-based traffic prediction methods are considerably smaller in scale compared to natural language processing and vision models. These baseline methods are generally capable of being trained within the required time-frame using limited GPU and memory resources, achieving real-time prediction.

Conclusion

This paper proposed a deep learning-based model TARGCN for traffic flow prediction. The TARGCN model fuses an Emb-GCN layer, a series of gated recurrent units, and a TA layer in a framework to exploit both dynamic spatial correlations between traffic nodes and temporal dependencies between time slices. The Emb-GCN layer is applied to extract the spatial correlation of traffic flow at a fine-grained level. Node embedding matrix and node parameter learning techniques are combined in this layer to learn the specific traffic pattern for each traffic node. A GRU layer stacked by a series of GRU cells is followed to extract traffic patterns

Table 4 Results of ablation experiments

Methods		TARGCN-linear	TARGCN-SA	TARGCN-noTA	TARGCN-noGate	TARGCN (proposed)
Dataset	Metrics					
PeMSD8	MAE	17.36	17.22	15.74	15.46	15.14
	RMSE	27.91	27.26	25.22	24.88	24.62
	MAPE (%)	11.46	10.79	10.15	10.04	9.90

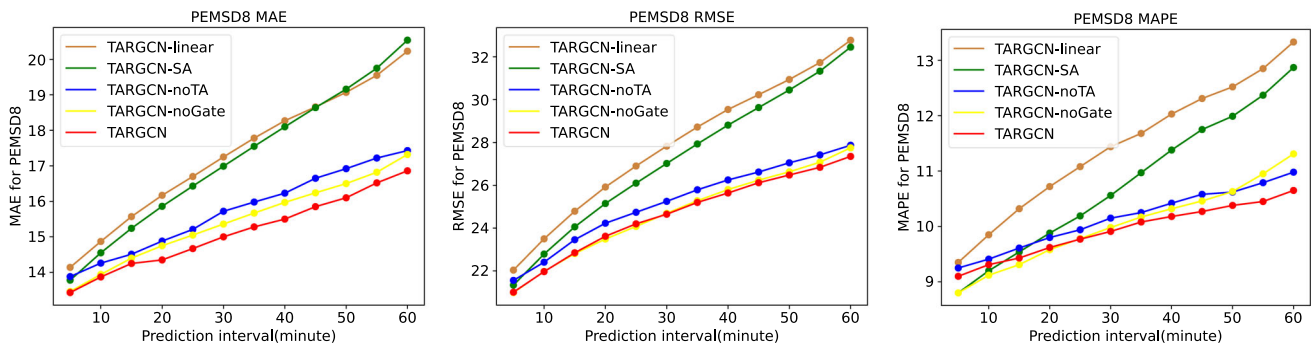


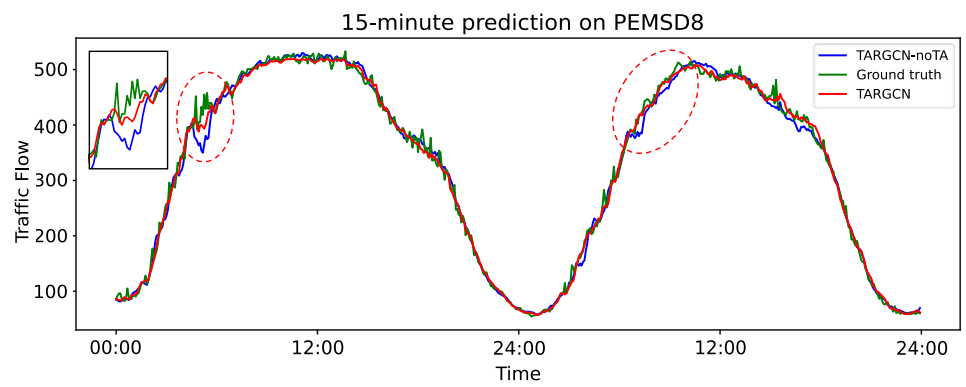
Fig. 8 Components analysis for ablation experiments

from the traffic flow of adjacent nodes in the past few time slices. This layer facilitates the model to capture both temporal and spatial correlations simultaneously. Subsequently, an attention layer is employed in the temporal dimension to extend the receptive field of GRU. This strategy enhances the ability of the proposed method for long-time prediction. Finally, a prediction layer using a 1D convolutional neural network is utilized to generate the predicted values. The fusion framework not only can fit the degree of interconnectedness between traffic nodes by traffic flow adaptively but also capture the specific traffic pattern for each node. This makes the model achieve promising prediction accuracy. Experiments on real-world traffic datasets demonstrate the advantage of TARGCN over compared state-of-the-art methods. Comparisons on average prediction errors indicate that the proposed method is basically superior to other methods. Especially for long-time predictions of 30 and 60 min, it exhibits a remarkable margin from the compared methods. Moreover, the effectiveness of each component of the proposed model is verified by the ablation experiments.

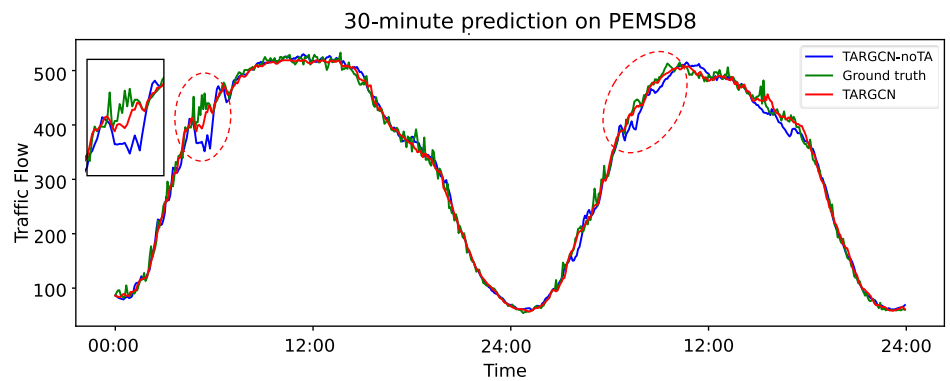
Still, the proposed TARGCN has some limitations and shortcomings. First of all, the model predicts the future flow from purely the historical flow while other factors, especially the weather, are not taken into account. This leads to the deficiency of robustness of the model. Although weather changes can reflect the changes in flow, this change transmis-

sion will be delayed. This results in a decline in the accuracy of the model over a period of time. The main reason we have not taken the weather into account is the difficulty of obtaining accurate real-time weather conditions. Inaccuracies of weather conditions, both past and future, may result in bad predictions. Alignment and multi-modal fusion for the weather and traffic data is also an open problem in the field. Actually, some works fusing the weather factor into the model did not achieve better predictions. In future work, we are more likely to consider a multi-stream style combined with rough weather conditions to improve prediction accuracy under variable weather conditions. Another weakness of the proposed model is its structure is based on RNN and a series of GRU cells in the GRU layer are sequentially connected which limits its ability to be parallelized. Since the proposed model requires to be retrained when the road network environment or traffic pattern changes, the serialization of the kernel calculation will affect the real-time performance of the system. Moreover, the proposed model applied the GRU fusion-based network and the learned adjacent matrix to make it more suitable for the freeway graph-based network pattern. This makes the model not advantageous on other types of road networks. Exploiting spatial-temporal dependencies between multi-adjacent nodes in the traffic to improve the adaptability to the grid-based road network for the model is another focus of our future work.

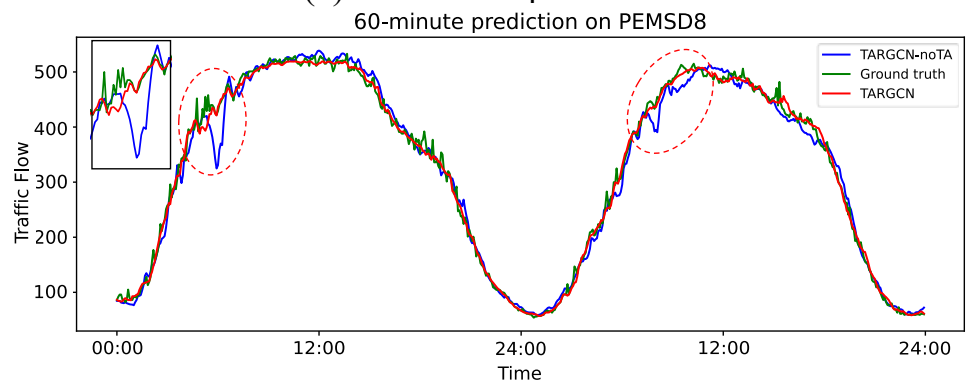
Fig. 9 Predicted traffic flows for the 127th node of PEMS8 in 24 h for TARGCN and TARGCN-noTA



(a) 15-minute prediction



(b) 30-minute prediction



(c) 60-minute prediction

Table 5 Number of parameters of methods

Methods	STGCN	DCRNN	ASTGCN	STSGCN	AGCRN	IGCRRN	STGMN	Proposed
Number of parameters	211,596	149,057	450,031	2,024,445	748,810	–	525,294	446,778

Acknowledgements This paper was supported in part by the National Natural Science Foundation of China under Grants 61772087, Natural Science Foundation of Hunan under Grants 2022JJ30621 and 2016JJ2005, Scientific Research Foundation of Hunan Provincial Education Department of China under Grants 19B004 and 16B006, Natural Science Foundation of Changsha under Grant No. KQ2202215.

Funding National Natural Science Foundation of China, 61772087, Yun Song, Natural Science Foundation of Hunan Province, 2022JJ30621, Yun Song.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

References

- Williams BM, Hoel LA (2003) Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results[J]. *J Transp Eng* 129(6):664–672
- Lu Z, Zhou C, Wu J et al (2016) Integrating granger causality and vector auto-regression for traffic prediction of largescale WLANs [J]. *KSII Trans Internet Inf Syst* 10(1):136–151
- Wu CH, Ho JM, Lee DT (2004) Travel-time prediction with support vector regression [J]. *IEEE Trans Intell Transp Syst* 5(4):276–281
- Van Lint JWC, Van Hinsbergen C (2012) Short-term traffic and travel time prediction models[J]. *Artificial Intelligence Applications to Critical Transportation Issues* 22(1):22–41
- Shi X, Chen Z, Wang H et al (2015) Convolutional LSTM network: A machine learning approach for precipitation nowcasting[J]. *Adv Neural Inf Process Syst* 28:802–810
- Zhang J, Zheng Y, Qi D. Deep spatio-temporal residual networks for citywide crowd flows prediction[C]. *AAAI conference on artificial intelligence (AAAI)*. 2017,31(1).
- Guo S, Lin Y, Li S et al (2019) Deep spatial-temporal 3D convolutional neural networks for traffic data forecasting[J]. *IEEE Trans Intell Transp Syst* 20(10):3913–3926
- Zhao L, Song Y, Zhang C et al (2019) T-GCN: A temporal graph convolutional network for traffic prediction[J]. *IEEE Trans Intell Transp Syst* 21(9):3848–3858
- Yaguang Li, Rose Yu, Cyrus Shahabi, et al. 2017. Diffusion convolutional recurrent neural network: Data-driven traffic forecasting [C]. *International Conference on Learning Representations (ICLR)*, 2018.
- Bai L, Yao L, Li C et al (2020) Adaptive graph convolutional recurrent network for traffic forecasting[J]. *Adv Neural Inf Process Syst* 33:17804–17815
- Guo S, Lin Y, Feng N et al (2019) Attention based spatial-temporal graph convolutional networks for traffic flow forecasting[C]. *AAAI conference on artificial intelligence (AAAI)* 33(01):922–929
- Hochreiter S, Bengio Y, Frasconi P, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies[J]. *Wiley-IEEE Press*, 2001: 237–243.
- Vaswani A, Shazeer N, Parmar N et al (2017) Attention is all you need[J]. *Adv Neural Inf Process Syst* 30:5998–6008
- Guo K, Hu Y, Sun Y et al (2021) Hierarchical graph convolution networks for traffic forecasting[C]. *AAAI Conference on Artificial Intelligence (AAAI)* 35(1):151–159
- He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*. 2016: 770–778.
- Lin Z, Feng J, Lu Z et al (2019) Deepstn+: Context-aware spatial-temporal neural network for crowd flow prediction in metropolis[C]. *AAAI conference on artificial intelligence (AAAI)* 33(01):1020–1027
- Yao H, Tang X, Wei H et al (2019) Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction[C]. *AAAI conference on artificial intelligence (AAAI)* 33(01):5668–5675
- Zhang J, Zheng Y, Sun J et al (2019) Flow prediction in spatio-temporal networks based on multitask deep learning[J]. *IEEE Trans Knowl Data Eng* 32(3):468–478
- Yu B, Yin H, and Zhu Z. Spatio-temporal graph convolutional networks: a deep learning framework for traffic forecasting[C]. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2018: 3634–3640.
- Geng X, Li Y, Wang L et al (2019) Spatiotemporal multi-graph convolution network for ride-hailing demand forecasting[C]. *AAAI conference on artificial intelligence (AAAI)* 33(01):3656–3663
- Defferrard M, Bresson X, Vandergheynst P (2016) Convolutional neural networks on graphs with fast localized spectral filtering[J]. *Adv Neural Inf Process Syst* 29:3844–3852
- Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. *arXiv preprint arXiv:1609.02907*, 2016.
- Zheng C, Fan X, Wang C et al (2020) GMAN: A graph multi-attention network for traffic prediction[C]. *AAAI Conference on Artificial Intelligence (AAAI)* 34(01):1234–1241
- Song C, Lin Y, Guo S et al (2020) Spatial-temporal synchronous graph convolutional networks: A new framework for spatial-temporal network data forecasting[C]. *AAAI Conference on Artificial Intelligence (AAAI)* 34(01):914–921
- Li M, Zhu Z (2021) Spatial-temporal fusion graph neural networks for traffic flow forecasting[C]. *AAAI conference on artificial intelligence (AAAI)* 35(5):4189–4196
- Pascanu R, Mikolov T and Bengio Y. On the difficulty of training recurrent neural networks[C]. *International conference on machine learning (ICML)*, 2013: 1310–1318.
- Yu F, Koltun V. Multi-scale context aggregation by dilated convolutions[J]. *arXiv preprint arXiv:1511.07122*, 2015.
- Wang Y, Jing C (2022) Spatiotemporal Graph Convolutional Network for Multi-Scale Traffic Forecasting[J]. *ISPRS Int J Geo Inf* 11(2):102
- Jiang J, Han C, Zhao W X, et al. Pdformer: Propagation delay-aware dynamic long-range transformer for traffic flow
- Ji J, Wang J, Huang C, et al. Spatio-temporal self-supervised learning for traffic flow prediction[C]//*Proceedings of the AAAI conference on artificial intelligence*. 2023, 37(4): 4356–4364.
- Song Y, Bai X, Fan W, et al. MSSTN: a multi-scale spatio-temporal network for traffic flow prediction[J]. *International Journal of Machine Learning and Cybernetics*, 2024: 1–15.

32. Zeng H, Cui Q, Huang X H, et al. STTD: spatial-temporal transformer with double recurrent graph convolutional cooperative network for traffic flow prediction[J]. *Cluster Computing*, 2024: 1–21.
33. Zhang Q, Yin C, Chen Y et al (2022) IGCRRN: Improved Graph Convolution Res-Recurrent Network for spatio-temporal dependence capturing and traffic flow prediction[J]. *Eng Appl Artif Intell* 114:105179
34. Chen C, Petty K, Skabardonis A et al (2001) Freeway performance measurement system: mining loop detector data[J]. *Transp Res Rec* 1748(1):96–102
35. Ni Q, Zhang M (2022) STGMN: A gated multi-graph convolutional network framework for traffic flow prediction[J]. *Appl Intell* 52:15026–15039
36. Xiong, L., Hu, Z., Yuan, X. *et al.* Multi-scale attention graph convolutional recurrent network for traffic forecasting[J]. *Cluster Comput* 2024,27, 3277–3291
37. Zhibo Xing, Mingxia Huang, Wentao Li, Dan Peng. Spatial linear transformer and temporal convolution network for traffic flow prediction[J]. *Scientific Reports*. 2024;14(1):1–12.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.